6-22-2011

The Hyper-log-chromaticity space for illuminant invariance

Samuel Brisebois

The Hyper-Log-Chromaticity Space for Illuminant Invariance.

by

Samuel Brisebois

BEng Computer Engineering, Royal Military College of Canada, 2005

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Chester F. Carlson Center for Imaging Science
Rochester Institute of Technology

June 22^{nd}, 2011

Signature of the Author _____

Accepted by _____
Coordinator, M.S. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Samuel Brisebois
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
M.S. degree in Imaging Science

Dr. David W. Messinger, Thesis Advisor

Dr. Michael Gartley

Dr. Emmett J. Ientilucci

Date

THESIS RELEASE PERMISSION
ROCHESTER INSTITUTE OF TECHNOLOGY
CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE

Title of Thesis:

The Hyper-Log-Chromaticity Space for Illuminant Invariance.

I, Samuel Brisebois, hereby grant permission to Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature _____

Date

The Hyper-Log-Chromaticity Space for Illuminant Invariance.

by

Samuel Brisebois

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Master of Science Degree
at the Rochester Institute of Technology

Abstract

Variation in illumination conditions through a scene is a common issue for classification, segmentation and recognition applications. Traffic monitoring and driver assistance systems have difficulty with the changing illumination conditions at night, throughout the day, with multiple sources (especially at night) and in the presence of shadows. The majority of existing algorithms for color constancy or shadow detection rely on multiple frames for comparison or to build a background model. The proposed approach uses a novel color space inspired by the Log-Chromaticity space and modifies the bilateral filter to equalize illumination across objects using a single frame. Neighboring pixels of the same color, but of different brightness, are assumed to be of the same object/material. The utility of the algorithm is studied over day and night simulated scenes of varying complexity. The objective is not to provide a product for visual inspection but rather an alternate image with fewer illumination related issues for other algorithms to process. The usefulness of the filter is demonstrated by applying two simple classifiers and comparing the class statistics. The hyper-log-chromaticity image and the filtered image both improve the quality of the classification relative to the un-processed image.

Acknowledgements

I acknowledge knowledge...

Thanks to:
Dr M. Gartley for helping me throughout my research, verifying my work and specially for providing the simulated images I used.

Cindy for taking care of everyone in the department and making sure we can all focus on our research.

My committee for leading me to my Master's degree.

I dedicate my research to my family and Justine who supported me and kept me going when I believed I would never finish this. For the first time in my life I had to focus on a single objective for a very long period of time and it wasn't easy for me. They were there to remind me why I was doing this and that I would be returning to multiple pressing jobs soon enough.

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Canadian Forces, the Department of National Defence, or the Government of Canada.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last century, advancements in electronics and circuitry have enabled the automation or improvement of a wide range of tasks and devices. Those advances have facilitated the work of the humans involved or removed them entirely from the process. This is not really the case of surveillance/monitoring activities. Some of these activities can be automated when the environment is highly predictable/stable. In controlled environments, objects of interest can be made obvious using high-contrast (black on white) or unique targets that could not possibly be found naturally in the environment. Machines can then be trained for those very specific/organized situations while being very efficient.

When it comes to applications with unpredictable objects of interest in changing environments (shown in Figure 1.1[1]), the human observer has yet to be replaced. Human vision allows observers to recognize, identify and track a wide variety of objects (including people) in changing conditions (background and illumination) very reliably. We do so using only shapes and colors in the visible spectrum. This kind of capability is required for surveillance and monitoring activities. The complexity of the task is immense. If the task of recognizing ground vehicles is considered, the task is to recognize motorcycles, cars, SUVs, trucks, vans, etc. which all have a multitude of brands, models, colors, designs that change throughout the years; really no simple task. Identifying humans is not any easier. We come in a wide range of sizes, shapes, and dress. We are not always positioned

[1] http://weburbanist.com/2007/08/10/interpreting-the-city-creative-urban-photography-at-three-different-scales/

Figure 1.1: Changes in illumination during a 24 hour period in an urban scene.

the same. This is not a simple task either.

Computer vision enables human observers to use modalities inaccessible to the human eye (radar, infrared, X-ray, etc.). Radar based Collision Avoidance System (CAS) have started to appear on consumer vehicles (note: they are more then likely using the radar as a range finder, not as an imager). Some of the research for a Pedestrian Detection System (PDS) uses infrared. Neither of these solutions is entirely satisfactory for surveillance. Radar is not really appropriate to detect humans. Infrared can help detection but is not exactly tailored for recognition. A hybrid of both would still not be sufficient to replace a human observer in decent visibility condition. There are also methods that focus on tracking specific features like pairs of headlights on vehicles, those methods will automatically fail when those features are not found (broken headlight, motorcycles rarely have two headlights and could be confused as a single object when they travel in pairs).

The objective of this research is to simplify images by reducing the variations due to illumination effects and shadows to improve application/task results. For this research to be relevant for the widest range of applications, it should have the least possible constraints. For this to be true, it will be assumed that very little information is known about the scene. The sensor spectral response will be the only known variable in the system, leaving the scene and illumination as unpredictable.

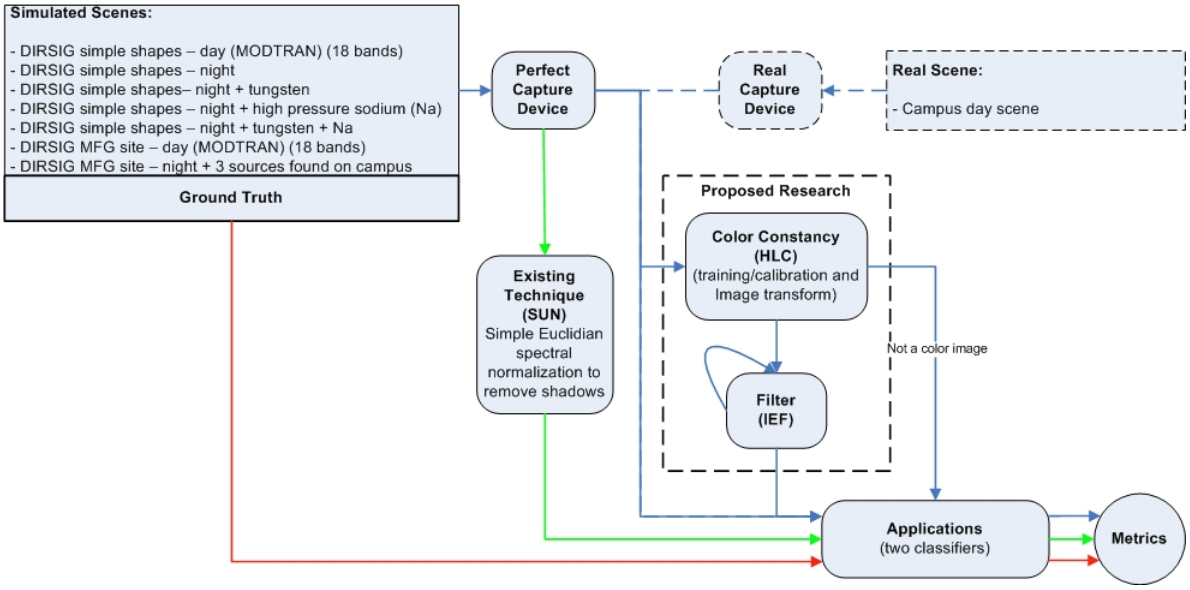


Figure 1.2: Objective Diagram.

1.1 Project Goal

This research will satisfy the following goals:

1. Provide an independent study of Finlayson's color constancy algorithm.
2. Provide a new chromaticity space for hyperspectral images less sensitive to illumination.
3. Provide a filter to reduce illumination effects (projected effects).
4. Provide an algorithm combining the chromaticity space and filter.
5. Test the new products with two classifiers on scenes rendered using DIRSIG with a variety of types of sources to determine optimal conditions to use them.

The following diagram (Figure 1.2) is an overview of this research as it was initially proposed:

Chapter 2

Background

This chapter contains information that is relevent to this research. It is a synthesis of the literature review made on various topics in regards to persistent surveillance at the beginning of this research that eventually led to the presented work. The sections are in order of relevance. The early sections are more generic and mostly included for completeness. The last sections are instrumental in the development of the proposed solution and its evaluation.

There are multiple sources of variations in an image, the most important for the purpose of this research, are the different materials in the scene. Since the objective is to be able to recognize/track objects based in the visible, it will require others sources of variation to be reduced or eliminated to ensure accuracy. An automated persistent surveillance system in the visible will be accurate if it can separate/identify all materials in the scene. The two main causes of difficulty in the visible are the sensor and the illumination. The specifications of the sensor will determine the information recorded from a scene. This is a short list of sensor specification which affect the image captured: pixel size, sensor type (material (silicon, germanium, etc.), build (CCD, CMOS, etc.)), size of the sensor array, noise, dynamic range, spectral response, optics (focal length, aperture), shutter speed, quantification, etc. "Applying an algorithm to images of actual scenes taken with a physical camera leads to the issues of camera characterization, image preprocessing, and the suitability of illuminant and reflectance sets used for algorithm calibration (training) to the images that will be encountered by the camera. Currently there is no satisfactory characterization of the images that an arbitrary vision system will encounter." [28] Therefore the present research establishes the

performance of algorithms mostly assuming a perfect sensor.

The illumination can cause additional variations in an image. The exact physics of interactions with light is not required in the presented research. The fact that the path taken by the light and the constitution of the medium in which it travels will influence it is known. As was mentioned before, the focus in this research is not on the illumination, but rather on the materials. The success of the research will be to remove observable illumination effects based on one captured image. The spectrum of the light source will change the apparent color of the material observed. This will only cause additional variations in material if different light sources are present, otherwise the color of the illumination does not impact our goal. Shadows will be covered in greater detail in the following section. They are caused by geometric effects. The other observable effect of illumination in a scene is the light falloff. Objects closer to a light source are brighter then objects further away. Variations in intensity can also be observed on a single object if it is big enough compared to the source (example: lit street at night, the asphalt directly under the street lamp receives more light then the stretch further away).

2.1 Shadow Removal

This section provides a simple description of shadows and an overview of shadow removal methods in the literature. It is important to note that most of the research on shadow removal does not achieve the goal of this research. Usually those methods refer to re-assigning pixels to the background of a scene for a segmentation task. They do not change the value of the pixel to remove the shadow, they only change its classification. The objective of the presented research is to change shadow pixels into well lit pixels so that segmentation (or other computer vision tasks) do not wrongfully assign pixels in the first place. This is more consistent with color constancy methods introduced in Section 2.4.

2.1.1 Shadows

Shadows are easily identified by a human observer, but it is not easy to have them recognized by computer vision. It is not merely the absence of light or a darker region in the scene; that would also describe dark objects. A shadow is an area where direct light from a light source cannot reach

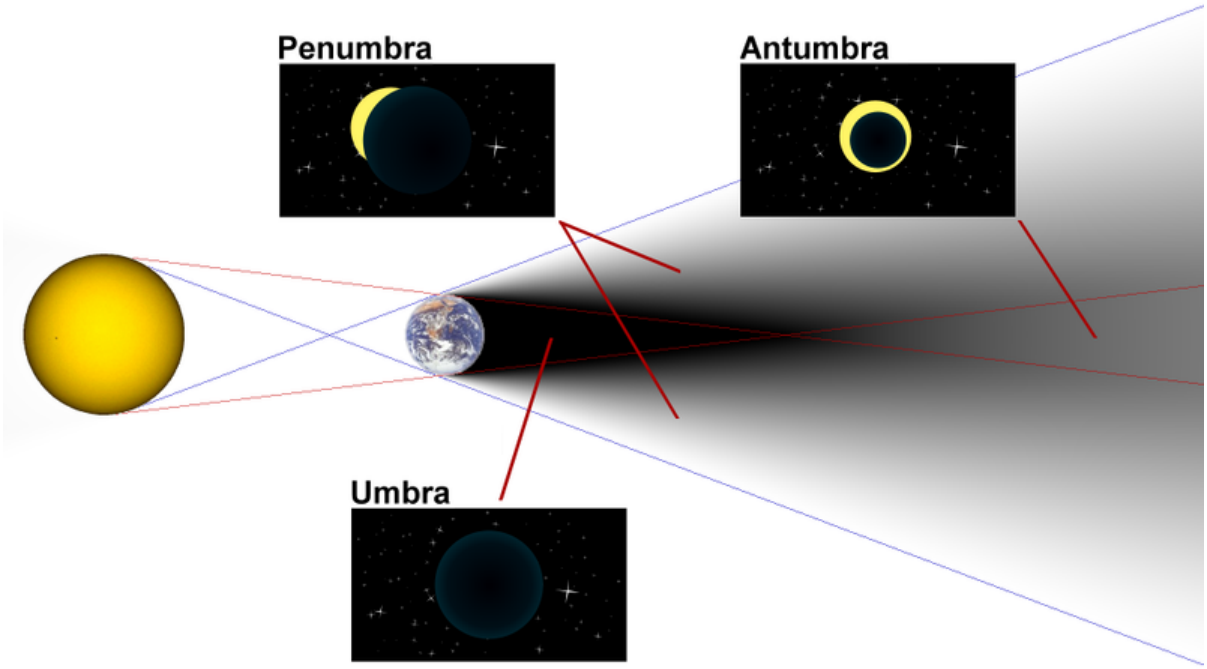


Figure 2.1: Celestial body nomenclature for shadows.

due to obstruction by an object. It occupies all of the space behind an opaque object with light in front of it. The cross section of a shadow is a two-dimensional silhouette, or reverse projection of the object blocking the light. Although this short description is accurate, it does not do justice to the complexity of a shadow in a scene and the issues it can cause to a visual system. A good starting point is the nomenclature used for celestial bodies. A shadow is split in three possible regions: penumbra, umbra and antumbra, as seen in Figure 2.1.

Figure 2.1 gives a representation of the regions in a simple scene. The penumbra is only present for broad sources, it receives a portion of light due to diffraction. The umbra is the region that receives no light from the source. The antumbra is the area where the penumbra from both side coexist. This demonstrates the difficulty of finding shadows based on changes of intensity alone. Another separation must be made between cast shadows versus self-shadows. The cast-shadow corresponds to the shadow shown in Figure 2.1. It is the area obstructed by another object. In a segmentation task, this part of the shadow belongs to the background. The self-shadow is the area

of an object blocked from light by itself. Contrary to the cast-shadow, this shadow belongs to the foreground in a segmentation task.

Unfortunately for computer vision, this is not a completely accurate model of the problem. A scene can have multiple sources with different spectra, in different locations, each creating shadow areas that interact with lit areas from other sources. This means that shadow regions can be interpreted as changes in intensity and in color. For instance, even day scenes have two sources. The sun is very directional and yellowish for the majority of the day. The sky on the other hand is omni-directional and light blue for the same period. The well lit areas of a day scene are illuminated by the combination of sunlight and skylight while shadows are only lit by skylight and are mostly blue in color.

Since a shadow is a reverse projection of the occluding object, it shares some of its spatial properties. In a way, a shadow follows the occluding object; this gives it the same trajectory, acceleration and speed in some cases. It will also share geometric properties with the occluding object. Depending on the lighting geometry, those shared properties could be strong features of the object (recognizable angle, profile) and could potentially confuse detection algorithms.

2.1.2 Algorithms

Much of the research in the literature is focused on the detection and tracking of moving objects for applications dealing with temporal image sequences (*ie* video). Shadows cause serious problems with segmenting and extracting moving objects due to the misclassification of shadow pixel as foreground. The background is the name given to all persistent elements found in a scene of interest while the foreground is the name for all elements that are new or do not belong in the scene. Shadows can cause objects to merge, change shape or the loss of objects. Prati *et al.* (2003) [2] present a taxonomy of shadow detection algorithms and a quantitative comparative study of four algorithms which provides significant insight.

"Deterministic approaches use an on/off decision process, whereas statistical approaches use probabilistic functions to describe the class membership." [2] Each approach has two subclassification, the deterministic approach decisions can be supported by model-based knowledge or not. Statistical approaches are parametric or not. All classes can rely on three types of features for shadow detection: spatial, spectral and temporal. There are different methods to build the back-

ground image. The most simple of the algorithms uses a single frame captured for this purpose. More complex methods use time varying backgrounds that use multiple frames to include new elements (see [27]). One method that appears often is the Gaussian Mixture Model (GMM) which is good for: a gradually changing background (like the gradual illumination), non-static background (like swinging leaves in the wind and television displays), and the sudden changes of the background. When the foreground objects are extracted by GMM algorithms, they usually include moving objects, cast-shadows and speckle noise [6].

Horprasert *et al.* (1999) [41] presented an example of a statistical nonparametric approach. "The distortion of the brightness α_i and the distortion of the chrominance CD_i of the difference between the expected color of a pixel and its value in the current image are computed and normalized with regard to their root mean square of pixel i. The values $\hat{\alpha}_i$, the thresholds (τ) and $\hat{CD_i}$ obtained are used to classify a pixel in four categories:

$$C(i) = \begin{cases} \text{Foreground:} & \hat{CD_i} > \tau_{CD} \text{ or } \hat{\alpha}_i < \tau_{\alpha lo} \\ \text{Background:} & \hat{\alpha}_i < \tau_{\alpha 1} \text{ and } \hat{\alpha}_i > \tau_{\alpha 2} \\ \text{Shadowed backg.:} & \hat{\alpha}_i < 0 \\ \text{Highlighted backg.:} & \text{otherwise} \end{cases} \tag{2.1}$$

The rationale used is that shadows have similar chromaticity, but lower brightness than the background model. A statistical learning procedure is used to automatically determine the appropriate thresholds." [2]

The statistical parametric approach used in Prati's *et al.* [2] study is from Mikic *et al.* (2000) [22]. "This algorithm claims to use two sources of information: local (based on the appearance of the pixel) and spatial (based on the assumption that objects and the shadows are compact regions). The *a posteriori* probabilities of belonging to background, foreground, and shadow classes are maximized. The *a priori* probabilities of a pixel belonging to a shadow are computed by assuming that $v = [R, G, B]^T$ is the value of the pixel not shadowed and by using an approximated linear transformation $\bar{v} = Dv$ (where $D = diag(d_R, d_G, d_B)$ is a diagonal matrix obtained by experimental evaluation) to estimate the color of the point covered by a shadow. The D matrix is assumed approximately constant over flat surfaces. If the background is not flat over the entire image, different D matrices must be computed for each flat subregion. The spatial information is

exploited by performing an iterative probabilistic relaxation to propagate neighborhood information. In this statistical parametric approach, the main drawback is the difficult process necessary to select the parameters. Manual segmentation of a certain number of frames has to be done to collect statistics and to compute the values of matrix D." [2]

"Choosing a model-based (deterministic) approach undoubtedly achieves the bests results, but is, most of the time, too complex and time consuming compared to the nonmodel-based. Moreover, the number and the complexity of the models increase rapidly if the aim is to deal with complex and cluttered environments with different lighting conditions, objects classes, and perspective views." [2] For these reasons, model-based method are usually only considered for indoor applications because the background and lighting can be controlled and kept constant (this also means they are predictable, the model can be built beforehand).

Many of the methods in the literature are based on the following deterministic nonmodel-based approach (by Cucchiara *et al.* (2001) [36]). It was determined that a shadow cast on a background does not significantly change the hue (H), reduces the saturation (S) and lowers the value (V) of a pixel. This relation is mathematically described as follows:

$$SP_k(x,y) = \begin{cases} 1 & \text{if } \alpha \leq \frac{I_k^V(x,y)}{B_k^V(x,y)} \leq \beta \wedge (I_k^S(x,y) - B_k^S(x,y) \leq \tau_S \wedge \mid I_k^H(x,y) - B_k^H(x,y) \mid \leq \tau_H \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

where $I_k(x,y)$ and $B_k(x,y)$ are the pixel values at coordinate (x,y) in the input image (frame k) and the background model (computed at frame k), respectively. The use of β prevents the identification as shadows those points where the background was slightly changed by noise, whereas α takes into account the "power" of the shadow.

The latest papers that refer to this approach get better results by further treating the results of Cucchiara *et al.* (2001) [36]. There are three recurring additions to the algorithm. (1) The algorithm is used multiple times with various thresholds to account for the variations in shadows (umbra and penumbra) or the shadow are further discriminated with a gradient based method. (2) Morphological operators are used to close the boundaries of shadow regions. (3) Connectivity algorithms are used to further detect shadows and discredit false alarms.

Another example of deterministic nonmodel-based approach by Stauder *et al.* (1999) [23] is based on three criteria: the presence of a darker uniform area, the presence of a high difference in luminance with regards to the reference frame, and the presence of static and moving edges. In addition, penumbra are detected by computing the width of each edge in the difference image. The penumbra causes a soft luminance transition that results in a wider edge in the difference image.

"The statistical approaches perform robustly in noisy data due to statistical modelling of noise. On the other hand, deterministic approaches (in particular, if pixel-based and almost unconstrained as Cucchiara *et al.* (2001) [36]) exhibit a good flexibility to different situations. Difficult sequences require, however, a more specialized and complete approach to achieve good accuracy." "If the environment is indoor, the statistical approaches are the more reliable since the scene is constant and a statistical description is very effective." [2]

The algorithms introduced in this section are too limited in scope and do not accomplish the goal of this research. Those methods could not be used for a sweeping camera or from a moving vehicle since the background would not be constant.

A deterministic nonmodel-based approach that relies on a single frame is presented by Chung *et al.* (2009) [30], it could therefore be used in more dynamic environments. Instead of searching for pixels that are nearly the same chromaticity and darker, they determine shadows by the ratio of hue over intensity. Unexpectedly, the metrics they use show excellent results. Hue is not linear in nature; technically the minimum and maximum value are similar colors. In this algorithm, if we consider the two extremum over the same intensity, one would be considered a shadow pixel while not the other. The quality of their results is presumed to be highly dependant on the nature of the scene. The images they show do sustain this argument. The majority of the images they use are mostly made of neutral colors. Dark objects in the scenes confuse their algorithm.

None of these methods offer a solution to retrieve the reflectance of the surface or provide an equivalent well lit value. In general, the outcome of an algorithm is simply to relabel the pixels identified as shadow to the background (see Figure 2.2). This way shadows are not passed-on to tracking algorithms or some other application where the focus is on the foreground. The few that mention an actual removal of shadow, do not go further than readjusting the luminosity of the pixel

(a) (b) (c) (d) (e)

Figure 2.2: Results from the Intelligent room sequence. Gray pixels identify foreground points and dark pixels indicate shadow points. (a) Raw image, (b) Statistical non parametric, (c) Statistical parametric, (d) Deterministic non-model 1, and (e) Deterministic non-model 2. [2].

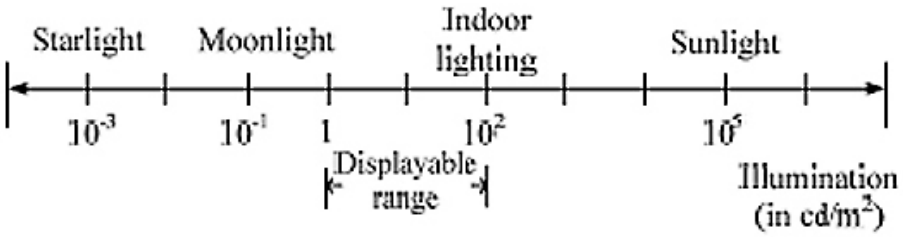


Figure 2.3: Scene luminance from [25].

which would create brighter blue regions in a day time image. All these methods are binary, either the pixel is under the brightest illuminant or the darkest. In a day scene this translates to which are lit by solar radiance and which are illuminated by the sky. Dark objects will often confuse these algorithms. Those algorithms would have difficulty with shadows within shadows (the shadow of a car in the shadow of an overpass) and would be inappropriate for night scenes. It could manage decent results for a moonlit scene without any artificial sources, the binary simplification would still be representative of the scene. Each added type of source would require an additional decision, this would be impractical for most surveillance applications.

2.2 Dynamic Range

Whether considering the world itself (scene) or sensors, the Dynamic Range (DR) is the ratio between the brightest and darkest illuminance value. Typically the world's dynamic range (see

Figure 2.3) exceeds the capacity of the sensors looking at it. The human eye manages to do well by having two detector types and three vision modes; scotopic, mesopic and photopic. Scotopic is the low light level (10^{-6} to 10^{-2} cd/m^2 = 40 dB) mode that only uses rods and is monochromatic. Photopic uses cones in normal lighting conditions (1 to 10^6 cd/m^2 = 60 dB) and is responsible for our color vision. Mesopic is an intermediate mode when lighting conditions are in between scotopic and photopic conditions. This allows objects to be seen under starlight or sunlight which represents a dynamic range of 90 dB. Only one mode can be used at a time and the transition from a mode to another can take some time. To put things in perspective, a typical consumer digital camera will capture at least 3 stops (9 dB); print film records 7 stops (21 dB); a computer LCD display will show 10 stops (30 dB). Luminance ratio, contrast ratio, EV difference and stops all relate to dynamic range:

$$1\_stop = 1EV\_difference = log_2(Contrast\_ratio). \tag{2.3}$$

Dynamic contrast ratio (another display value) on the other hand cannot be compared directly to DR (multi-modal like the human eye).The lux is the SI unit of illuminance and luminous emittance. It is used in photometry as a measure of the intensity, as perceived by the human eye. Because a candela [cd] is weighted by the spectral response of the eye, knowledge of the wavelengths observed is required in order to transfer to radiometric units (physical power [W/m^2]):

$$1lux[lx] = 1lm/m^2 = 1\frac{cd * sr}{m^2}. \tag{2.4}$$

In the context of imaging sensors, the dynamic range refers to the minimum and maximum radiance it can sense. The minimum is normally defined by the radiance required to generate a signal equal to the dark noise. The source of noise responsible for the lower limit depends of the sensor construct and operation (temperature is a major factor). For most sensors, the maximum is defined by the radiance necessary to fill up the capacity well (saturate). The dynamic range can be calculated using photons, radiance, illuminance or signal (amps or volts). Typically, the dynamic range is expressed in decibels (dB) following the equation:

$$DR = 20log_{10}\frac{S_{max}}{S_{min}}[dB]. \tag{2.5}$$

This definition has the advantage of reducing the values due to the properties of the logarithm and allows direct comparison of different sensors. Unfortunately it gives no indication of the general situation of the range; so that two sensors with the same range could be incapable of imaging the same scene (e.q. sensor A $S_{min} = 1$, $S_{max} = 10$, $DR_A = 20$; sensor B $S_{min} = 10$, $S_{max} = 100$, $DR_B = 20$). The factor of 2 is not present in the photographic definition of dynamic range but it is present in digital imaging wether the ratio is squared or not. Decibels are devised to represent power ratios, in some applications the square of the amplitude is proportional to power. Due to the properties of logarithms, multiplying by two the log of the ratio is the same as the log of the squared ratio. The result will be the same for the application in question if the proportionality between power and amplitude assumption holds. Hence dynamic ranges between photography and digital imaging must be compared carefully; the numbers provided for the human visual system were not multiplied by 2.

2.2.1 HDR Imaging

Only the regions from a scene that fit within the DR of a sensor will provide useful information. Regions too dark will be all zeros and regions too bright will all have the maximum value. Variations in brightness beyond the DR of a sensor are not observable, hence information is lost. The appellation HDR (High Dynamic Range or Wide Dynamic Range (WDR)) is used whenever a device/scene exceeds the DR of the accepted normal for the presently available technology. A clear definition or standard was not found in the literature. DR should be considered an important factor when recording images which have to be balanced with the other major parameters: spatial resolution and spectral resolution. An HDR system would be better suited for persistent surveillance then traditional systems. This is easily justified by looking at Figure 2.3 which shows the complete range of possible luminosity throughout a 24 hour period. A traditional system would require an automation of the exposure setting to be able to see the subject of interest and the usable information would be limited to the subject. An HDR system might still require auto-exposure but it would not be updated as often and more than the subject of interest would have information.

There are two principal areas of research in HDR imaging: research to capture/create HDR

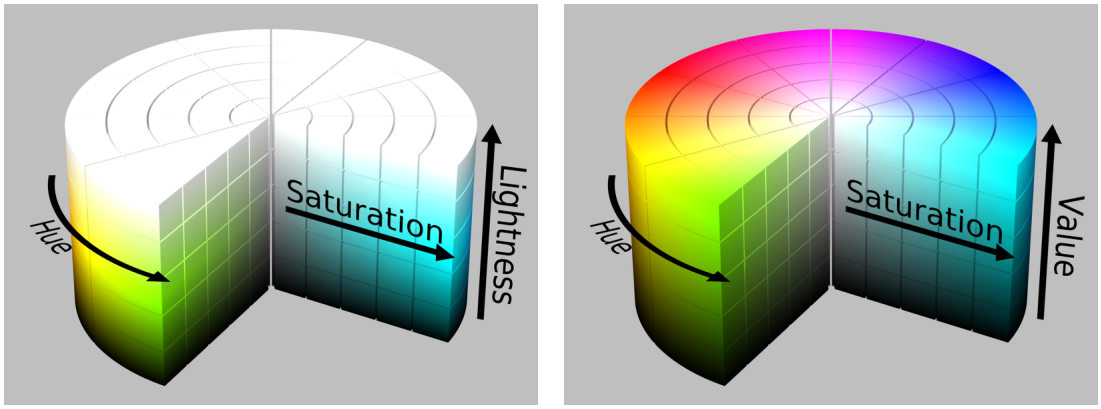
images and research to display it. Presently, most HDR images are an image product built from multiple Low Dynamic Range (LDR) exposures. This method is not particularly practical when dealing with moving sensors and/or objects. Work is being done to develop sensors capable of recording HDR images in the time of a single exposure. Noriko Ide *et al.* [34] present a prototype with interesting properties. Their sensor has a DR of 180 dB. Although it is only one example and still a prototype, the study of the impact of DR on the proposed algorithm seemed less relevant and the focus was put on other limiting aspects. This research has been conducted using the full unquantified range of data available.

2.3 Color Space

Color spaces are mathematical devices to describe the visible spectrum (from 380 nm to 720 nm). The most classical color spaces can be grouped into four main families [7], namely:

- The primary spaces which are based on the trichromatic theory, assuming that it is possible to match any color by mixing appropriate amounts of three primary colors.
- The luminance-chrominance spaces where one component represents the luminance and the two others the chrominance.
- The perceptual spaces which try to quantify the subjective human color perception using the intensity, the hue and the saturation.
- The independent axis spaces resulting from different statistical methods which provide as few correlated components as possible.

The color space used by an algorithm will impact its performance. RGB is the most popular color space, it stands for red, green and blue (the primary colors) which are the three bands recorded by most imaging devices. The exact composition of these bands depends on the sensor (usually the sensor's filter), generally speaking the shorter wavelengths contribute to the blue band (approximately 380nm to 490nm), the center of the visible form the green band (approx. 490nm to 600nm) and the rest of the spectrum result in the red channel. This is the most common color space and it is the most familiar, as it is used in education (how to obtain different colors with



(a) HSL (b) HSV

Figure 2.4: Two examples of luminance-chromaticity space.

paints). YCM (yellow, cyan and magenta) is another primary space referred to as secondary colors but it is mainly used by the impression industry. Although most devices record images in a primary space, very few shadow removal algorithms use it (for instance most methods introduced above use chromaticity-luminance spaces). This results in a wide variety of transforms to go from a primary color space to other spaces.

The main attractive feature of luminance-chrominance spaces is their correspondence to the human perception of colour. It allows human descriptions to hold meaning in algorithms. 'Darker' is a reduction of the luminosity (L (for luminance, luminosity or lightness) or I (for intensity) or V (for value)) channel which would be a reduction of all three channels in RGB. A change of color results in a different hue (H) instead of a different mixture of primary colors and the transition from vivid to faded colours is a change in saturation (S). The following Figure (2.4[1]) shows two possible representations of luminance-chrominance space (hue and saturation form the chrominance).

The transforms from RGB to luminance-chromaticity spaces are not standardized, the literature reveals they may vary both in relation and domain. Care must be taken when comparing images in those color spaces, the comparison holds little value if different transforms were used. Typically, those color spaces are based on a cylindrical coordinate system (θ, r, h). The hue is represented by

[1] http://en.wikipedia.org/wiki/HSL_and_HSV

θ, the saturation by r and the luminance by h (as shown in Figure 2.4). Normally the color spaces are limited to eight bits so the hue (0 to 360°) is quantized to [0,255], the domain of the saturation and luminance vary too much to mention a norm. The following equations are an example of an RGB to HSL transform:

$$\begin{aligned} V_1 &= min\{R, G, B\} \\ V_2 &= max\{R, G, B\} \\ L &= \frac{1}{2}(minV + maxV) \\ V_3 &= min\{L, 1 - L\} \\ S &= \frac{\frac{1}{2}(maxV - minV)}{V_3} \end{aligned}$$

where $maxV$ and $minV$ are respectively the maximum and minimum value found in the image. If R is the maximum value:

$$H = \frac{G - B}{V_2 - V_1}$$

If G is the maximum value:

$$H = 2 + \frac{B - R}{V_2 - V_1}$$

If B is the maximum value:

$$H = 4 + \frac{R - G}{V_2 - V_1}$$

And finally if the resulting H from the previous equations is negative ($H < 0$) then:

$$H = 60 * H + 360$$

This transform is implemented by the Matlab function "colorspace.m" available from www.mathworks.com (it contains the transforms for 13 color spaces) [20]. Many transforms, like the one shown here, are based on the minima and maxima found in an image. This means those color spaces may complicate image comparison unless the images are from the same scene with little to no change in illumination and the sensor noise is low. Those transforms can also have an effect (compression/stretching) on the DR of an image. Users must be careful when comparing the hue information because of its 'circular' nature, namely zero and 360° are actually the same hue. Neutral colors (very little hue) like black, white, gray, beiges, etc. are all mapped near the center, so different colors can

be easily confused in this region. Black, white and all levels of gray are only distinguished by the third dimension.

The perceptual family of color spaces are similar in format to luminance-chromaticity spaces, but their transforms have an important difference. The human visual response and various psycho-visual factors are considered. The standard is established by the *Commission Internationale de l'Éclairage(CIE)*, which is where the spaces take their name from: CIE XYZ, CIE L*,a*,b* (or CIELab), etc. The spaces evolve as studies permit a better understanding of the human perception of color and the CIE manages to incorporate color appearance phenomena. Those color spaces are particularly useful when the output is for human observers. On the other hand, they would not be suited for this research because using these would lend the same limit (passing from hyperspectral data to 3 bands in the visible) than human observers to computer vision.

The last color space family is the result of the exploration of different means to record the visible spectrum (hyperspectral) or other portions of the spectrum (infrared). The output of these sensors is not in RGB, although experts can understand and interpret the data recorded, image products are always preferred. A simple method to reduce the dimensionality of the data set is to calculate the singular value decomposition (SVD) or the principal components analysis (PCA) of the image cube and attribute the top three dimensions to RGB channels for display [39]. This method is appropriate as long as the variance in the data set is correlated to information. Custom color mapping functions are made when it is not the case (i.e. variation in temperature recoded by an infrared imager; blue is cool, red is hot).

For the most part, images represented in these spaces supply no more data to the computer [7]. Each color space is a different representation of the same data which makes each of them better suited for different applications. As an example of this, Kumar *et al.*(2002) [35] tested which color space amongst RGB, XYZ, YC_rC_b (a luminance-chromaticity space designed for video pipelines (TV)), HSV and rgb (normalized RGB) was the best suited for shadow removal. They concluded that YC_rC_b was the best suited for the deterministic nonmodel-based algorithm tested. YC_rC_b is robust to brightly lit background and able to detect shadows quite precisely [35]. This demonstrates the color space used by an algorithm will impact its performance even if it provides the same

information.

2.4 Color Constancy

This section covers the algorithms that are not used in the presented research. The objective is not to convey a precise understanding of the various algorithms, but rather a generic sense of how they proceed and the relative quality of the results. As such, most of this section is a collection of quotes from various papers describing their method or conducting studies to determine which is the best approach.

"The light reaching our eye is a function of surface reflectance and illuminant color. Yet the colors that we perceive depend almost exclusively on surface reflectance; the dependency that is due to illuminant color is removed through color constancy computation. As an example, the white page of a book looks white whether viewed under blue sky or under artificial light. However, the processes through which color constancy is attained are not well understood. Indeed, the performance of color constancy algorithms in computer vision remains quite limited." [13] Color constancy is the ability to determine the colors of objects irrespective of the illumination conditions and of the nearby objects color [24]. Although color constancy research is rarely used for this purpose, that ability would actually eliminate shadows (not just remove shadows from the foreground as the algorithms covered in Section 2.1.2). As it was discussed in a previous section, the errors in segmentation or classification are due to the differences in illumination, by estimating the surface reflectance irrespective of illumination, the source of errors is removed.

There are many different approaches to color constancy, some will be briefly introduced in this section: Retinex [E.H. Land (1977) [31]], Gray-World (GW) [G. Buchsbaum (1980) [3]], Gamut-mapping, a neural network, and two statistical methods. Depending of their approach, the algorithms might be labeled "color invariant", "illuminant invariant", etc. All those names are tied to the same concept that seeks to estimate one of the two unknowns in sensor reaching irradiance equation (generally speaking, a valid estimate of one of the unknowns will allow one to estimate the other). It is argued that because a bright light striking a dark surface results in the same physical spectra as those of a dim light incident on a light surface, the magnitude of RGBs cannot be recovered. Consequently, color constancy algorithms (introduced below) attempt

only to recover image chromaticities under the reference light (canonical illuminant): They solve a two-dimensional problem [13]. The goal of computational color constancy is to account for the effect of the illuminant, either by directly mapping the image to a standardized illuminant invariant representation, or by determining a description of the illuminant which can be used for subsequent color correction of the image. Other algorithms estimate the color of the illuminant as defined by the camera response to a pure white, or a projection thereof (chromaticity) [29].

The correction of chromaticity is often sufficient because an illuminant magnitude is often implicitly present. For example, when a picture is taken, either a human operator or some mechanism has often set the aperture to a reasonable value. Thus, a correction for chromaticity, which leaves the overall brightness the same, is often sufficient for image reproduction applications [29]. On the other hand, it is commonly recognized that specular highlights carry information about the illuminant chromaticity [Shafer(1985) [40],Richard(1995) [37]], and the fact that they are relatively bright is of use to some algorithms [29]. In the case of illuminant chromaticity, the assumption of spatial uniformity is generally more valid then for surface reflectance chromaticity. For this reason, many of the approaches are developed to deal with a single illuminant in the scene. However, authors are aware that inter-reflection can still cause major deviations and, in natural images, there are often multiple sources of illumination. For example, in outdoor images the sun and the sky often illuminate different parts of a scene with varying strengths. Color constancy under such conditions is often beyond the scope of algorithms [28].

Central to solving the color constancy problem is recovering an estimate of the scene illumination and it is that problem which is normally the focus of algorithms. Part of the difficulty is due to the fact that the problem is inextricably tied up with other confounding phenomena. Algorithms must account for changes in image intensity and color which are due to the shape of the objects, viewing and illumination geometry, as well as those due to changes in the spectral power distribution of the illuminant ($E(\lambda)$) and the spectral reflectance properties ($S(\lambda)$,σ) of the imaged objects. Thus, to simplify the problem, many researchers [Land(1977) [31],Maloney *et al.*(1986) [32],Forsyth(1990) [10],D'Zmura *et al.*(1994) [33]] have considered a simplified two-dimensional world in which all objects are flat, matte, Lambertian surfaces, uniformly illuminated. The task for a color constancy algorithm is to transform the pixel value so that they become independent of $E(\lambda)$ and, hence, correlate with $S(\lambda)$. Equivalently, the problem can be posed as that of recovering an estimate of $E(\lambda)$ since with this knowledge, it is relatively straightforward

to recover an image which is independent of the prevailing illumination. Fortunately, it is often unnecessary to recover the full spectra of lights and surfaces, rather it is sufficient to represent a light by the response of a device to a perfect diffuser viewed under it and, similarly, to represent a surface by the response it induces under some canonical light [15].

Authors recognize that the problem does not have a unique solution, they try to exploit information in the image to recover the most likely solution. They pose the problem in a probabilistic framework and Sapiro(1985) [38] has developed an algorithm based on the Probabilistic Hough Transform. The neural network approach [Funt *et al*(1996) [4]] to color constancy can similarly be seen as a method of dealing with the inherent uncertainty in the problem. While these algorithms which model and work with uncertainty represent an improvement over earlier attempts at solving the color constancy problem, none of them can be considered the definitive solution [15].

Retinex

Retinex [Land (1964) [31]] is the first computational model to explain and achieve color constancy. The emphasis of Retinex theory is on human vision, and goes beyond simple illuminant estimation. Hence, computational color constancy algorithms emerge from Retinex more as a process of analogy than through specification by the original researchers [29]. Land and McCann Retinex theory is based on psychophysical experiments using Mondrian patterns (geometric paintings made popular by Piet Mondrian [1872-1944]). These experiments supported the existence of a quantity named lightness which was associated to the objects of the scene regardless of changes in the illumination or in the position of the objects in the scene. These experiments show that the lightness information is processed independently in the three sets of color receptors. In the past thirty years, numerous Retinex implementations have been published and effort has been made to improve the original Retinex algorithm.[Brainard *et al.*(1986), Provenzi *et al.*(2007)] The lightness information is estimated by computing sequential ratios between values at adjacent points of a series of random paths in the image. Changes above a certain threshold are considered as changes in reflectance. If instead color changes are smaller than the threshold they are considered as illumination changes [24].

Even though it is the oldest method, it is one of the most computationally demanding algorithms in the literature. Many of the offsprings of this methods are focused on reducing the computational

load rather then getting better results. Technically, the Retinex theory is not suited to remove hard shadows as the large change would be considered a change in reflectance instead of illumination. The following paragraph describes a variant of Retinex that was used in a two part study of various algorithms [28, 29].

"The SCALE-BY-MAX algorithm estimates the illuminant by the maximum response in each channel. It is a limiting case of one version of Retinex [McCann *et al.*(1976), Brainanrd *et al.*(1986)]." [29] K. Barnard *et al.* [29] also remark on the sensitivity to the dynamic range of this method. If specularities are not clipped, this method provides an excellent estimate of illuminant chromaticity.

Gray-World

The so-called Gray-World (GW) [Buchsbaum(1980) [3]] algorithm has been around for some time also, but it is one of the simplest and it is still widely used [15]. The GW method is based on the assumption that the spatial average of surface reflectances in a scene is achromatic. Since the light reflected from an achromatic surface is changed equally at all wavelengths, it follows that the spatial average of the light leaving the scene will be the color of the incident illumination. In other words, it assumes that the average of the observed image is a good estimate of the camera response to gray. Ideally gray is defined by the expected average over the application domain. This is not generally available, so algorithms are based on alternatives. Some of them use a 50% uniform reflectance for gray,others use the average of the reflectance spectra in the reflectance dataset [28]. While it is often used for color constancy, the GW algorithm has a number of limitations. First, Gershon *et al.* (1988) have pointed out that the spatial average is biased towards surfaces of large spatial extent. They proposed a modified algorithm which alleviates this problem by segmenting the image into patches of uniform color prior to estimating the illuminant [15]. A second limitation of the GW algorithm is highlighted by the equivalency equation in which the identity matrix does not accurately represent the knowledge about the interaction of lights and surfaces. Improving color constancy then, amounts to finding matrices which more accurately encode that knowledge [15].

Gamut-Mapping

Two researchers related to Gamut-Mapping (GM) methods seem prominent: Forsyth and Finlayson. Those methods are the remote-sensing equivalent of brute force approaches based on look-up tables. Those method do not attempt to find a unique solution to the problem; rather the set of all possible solutions are calculated and, from this set, the best solution is chosen [15].

There are a number of algorithms based on Forsyths (1990) [10] GM approach. The first step of Forsyth's approach (called CRULE for "coefficient rule") is to form the set of all possible RGB due to surfaces in the world under a known, canonical illuminant. Forsyth showed that color gamuts are closed, convex, bounded, and that, most importantly, each is a strict subset of the set of possible image colors. The gamut of possible image colors for a light can be determined by imaging all possible surfaces (or a representative subset thereof) under that light [15]. This set is convex and it is represented by its convex hull. Similarly, the set of all possible RGB under the unknown illuminant is represented by its (unknown) convex hull. Under the diagonal assumption of illumination change, these two hulls are a unique diagonal mapping (a simple 3-D stretch) of each other. The goal is to estimate that diagonal mapping [29]. Finlaysons (1996) [9] Color in Perspective algorithm adds two additional ideas. First, the gamut-mapping method can be used in the chromaticity space. Second, the diagonal maps can be further constrained by restricting them to ones corresponding to expected illuminants [28].

Another similar approach, Color by Correlation (another method by Finlayson *et al.* (2001) [15]), is to pre-compute a correlation matrix which describes the extent to which proposed illuminants are compatible with the occurrence of image chromaticities. Each row in the matrix corresponds to a different training illuminant. The matrix columns correspond to possible chromaticity ranges resulting from a discretization of space, ordered in any convenient manner [29]. In the second version of Color by Correlation, the correlation matrix is set up to compute the probability that the observed chromaticities are due to each of the training illuminants [28]. Since rather than saying that an illuminant is possible if and only if it is consistent with all image colors, Color by Correlation instead looks for illuminants that are consistent with most image colors. This subtle change cannot be implemented into the CRULE algorithm, if no illuminant is globally consistent, there is no solution to color constancy [15].

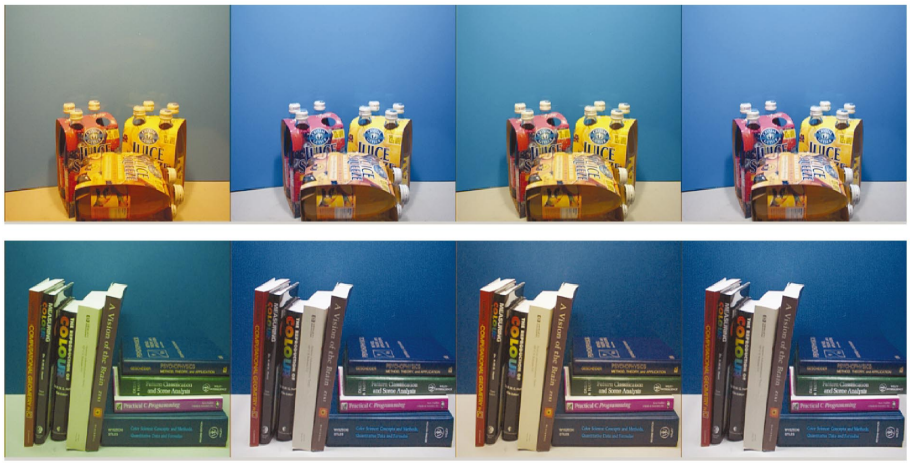


Figure 2.5: Gamut mapping examples from [15].

Figure 2.5 is from Finlayson's paper on color by correlation [15], it compares the original image (first from the left) to the correction based on the measured illuminant (second from the left), a 2D gamut mapping (second from the right) and his method (first on the right). It can be observed that their method is visibly very similar to the canonical illuminant. On the other hand, the shadows are still present.

There are several potential problems with the basic GM methods as described. First, due to noise, and other sources of mismatches between the model and the real world, an observed set of chromaticities can yield zero probability for all illuminants, even if the illuminant, or a similar one, is in the training set. Second, the illumination may be a combination of two illuminants, such as an arbitrary mix of direct sunlight and blue sky, and ideally we would like the method to give an intermediate answer [29]. Another problem with gamut mapping is that not all chromaticities correspond to plausible illuminants (for example, purple lights do not occur in practice). This observation is also simple to implement since we can simply restrict the columns to those corresponding to plausible lights [15].

Others

Retinex, Gray-World and Gamut Mapping seem like the most relevant work in the literature. The next approaches are just mentioned in passing to show what other avenues are being explored.

Funt *et al.* (1996) [4] uses a neural network trained to estimate the color of the illuminant. The neural network is a multilayer Perceptron with two hidden layers. The output signal from the two output neurones are real valued, and correspond to an estimate of the chromaticity of the scene illuminant [29].

"Sapiro (1985) [38] has recently proposed an algorithm for estimating the scene illuminant which is based on the Probabilistic Hough Transform. In this work, Sapiro represents lights and surfaces as low-dimensional linear models and defines, according to this model, a probability distribution from which surfaces are drawn. Given a sensor response from an image, a surface is selected according to the defined distribution. This surface, together with the sensor response, is used to recover an illuminant. If the recovered illuminant is a feasible illuminant (in Sapiro's case an illuminant on the daylight locus), a vote is cast for that illuminant." [15] Exploiting detailed image statistics requires a good match between the statistics used for calibration and the statistics the vision system encounters in the world. Ensuring a good match requires better characterization of image statistics than currently available [28].

"Brainard and Freeman (1997) [8] have given a Bayesian formulation of the color constancy problem. Their approach is again founded on a linear model representation of lights and surfaces. That is, each light and surface is represented by a weighted sum of a small number of basis functions so that these weights are sufficient to define a light or surface. Principal component analyses of collections of surfaces and illuminants were used to determine suitable basis functions and the corresponding weights for each light and surface. The authors then defined probability distributions for these weights and used Bayesian decision theory to recover estimates of the weights for the surfaces and illuminant in an image." [15]

2.5 Log-Chromaticity Color Space

Graham D. Finlayson published a number of articles with various co-authors (Steven D. Hordley, Paul M. Hubel, Mark S. Drew, Cheng Lu,and Brian V. Funt) showing their work in color constancy. Their work is the an important part of this research and will be referred to often. For the sake of brevity, 'Fin' (for Finlayson, the common author to all articles) followed by the year will be used when referring to those papers. In one of the earliest [13]) when they introduce this particular

color constancy algorithm. In the later publications, they introduced a brute force approach to re-orient their color space for images of unknown provenance and an algorithm to create shadow free images. Their paper, in the International Journal for Computer Vision [12], does an adequate summary of the progression of their work.

Fin's (2001) method is devised for the recovery of an illuminant invariant image from a three band (RGB) color image. The invariant image is independent of lighting, and also has shading removed: it forms a type of intrinsic image, independent of illumination conditions. While the essential definition of an intrinsic image is one that captures full reflectance information (Barrow and Tenenbaum 1978 [16]), Fin (2001) claim only to capture chromaticity information not full reflectance. Although shadow removal is not always perfect, the effect of shadows is so greatly attenuated such that many algorithms might benefit from the new method. The method devised finds an intrinsic reflectivity image motivated by the assumptions of Lambertian reflectance, approximately Planckian lighting, and fairly narrowband camera sensors. Nevertheless, the method still works well when these assumptions do not hold. Under those simplifying assumptions, the values of the logarithms of chromaticity band-ratios for pixels from the same surface, but under different (Planckian) lighting fall on a straight line; and every such line, for different surfaces, has the same slope. Projection orthogonal to this special direction results in a 1D (greyscale) invariant image that has shadows approximately removed. Originally, the invariant direction was gleaned via a preliminary calibration routine, using the camera involved to capture images of a color target under different lights. Subsequently, it was shown in principle [11] that the method works without the calibration step; the invariant direction is found for each specific image independently. Drew (2003) proposed the 2D chromaticity version of their method. In the most recent paper, Fin (2009) transition from the Shannon's entropy definition to quadratic entropy as the first definition can be quite sensitive to bin-width (refers to histogram theory, the size of the intervals (bin) used to count the pixel values).

This section shows how linear behaviour with lighting change results from the assumptions of Planckian lighting, Lambertian surfaces, and a narrowband camera. Consider the RGB color $\mathbf{R}$ formed at a pixel, for illumination with spectral power distribution $E(\lambda)$ impinging on a surface with surface spectral reflectance function $S(\lambda)$. If the three camera sensor sensitivity functions form a set $\mathbf{Q}(\lambda)$, then we have:

$$R_w = \sigma \int Q_w(\lambda) S(\lambda) E(\lambda) d\lambda \tag{2.6}$$

where w are the image channels and σ is Lambertian shading: surface normal dotted into illumination direction. If the camera sensor $\mathbf{Q}(\lambda)$ is exactly a Dirac delta function $Q_w(\lambda) = q_w \delta(\lambda - \lambda_w)$, then Equation (2.6) becomes

$$R_w = \sigma q_w S(\lambda_w) E(\lambda_w). \tag{2.7}$$

They then suppose the lighting can be approximated by Planck's law,

$$E_P(\lambda, T) = I \frac{2\pi h c^2}{\lambda^5} \left[e^{\frac{hc}{k\lambda T - 1}} \right]^{-1} \tag{2.8}$$

where I gives the overall light intensity, h is Planck's constant (6.626068E-34 Joule seconds), c is the speed of light in vacuum (299792458 m/s), k is Boltzmann's constant (1.381E-23 Joules/Kelvin), λ is the wavelength in meters, and T is the temperature of the blackbody radiator in Kelvin. For a wavelength of 380 to 720 nanometers and for blackbodies of 2000 Kelvin to 10000 Kelvin, Wien's approximation is valid and easier to use in the next steps. Then the spectral irradiance is expressed as:

$$E_W(\lambda, T) = I \frac{2\pi h c^2}{\lambda^5} e^{\frac{-hc}{k\lambda T}}. \tag{2.9}$$

The RGB color from Equation 2.7 is

$$R_w = \sigma q_w S(\lambda_w) I \frac{2\pi h c^2}{\lambda_w^5} e^{\frac{-hc}{k\lambda_w T}}. \tag{2.10}$$

Band ratios are formed with the geometric mean which effectively removes intensity and shading information such that

$$r_w = \frac{\sigma q_w S(\lambda_w) I \frac{2\pi h c^2}{\lambda_w^5} e^{\frac{-hc}{k\lambda_w T}}}{\sqrt[3]{\prod_{i=1}^{3} \sigma q_i S(\lambda_i) I \frac{2\pi h c^2}{\lambda_i^5} e^{\frac{-hc}{k\lambda_i T}}}} = \frac{\frac{q_w S(\lambda_w)}{\lambda_w^5} e^{\frac{-hc}{k\lambda_w T}}}{\frac{q_{gm} S(\lambda_{gm})}{\lambda_{gm}^5} e^{\frac{-hc}{k\lambda_{gm} T}}} \tag{2.11}$$

where the subscript gm indicates the geometric mean ($y_{gm} \equiv \sqrt[z]{\prod_{i=1}^{z} y}$) and r_w is the ratio of R_w with its geometric mean. The geometric mean of chromaticities is used instead of simple band ratios (as in Fin (2001) basic demonstration reproduced in appendix A), in order to not favor

one particular color channel. The lighting and surface reflectance information is separated by calculating the natural logarithm of the ratios as

$$\rho_w \equiv ln\left(r_w\right) = ln\left(\frac{\frac{q_w S(\lambda_w)}{\lambda_w^5}}{\frac{q_{gm} S(\lambda_{gm})}{\lambda_{gm}^5}}\right) + \frac{hc}{kT}\left[\frac{1}{\lambda_{gm}} - \frac{1}{\lambda_w}\right] = ln\left(\frac{s_w}{s_{gm}}\right) + (e_w - e_{gm})/T. \tag{2.12}$$

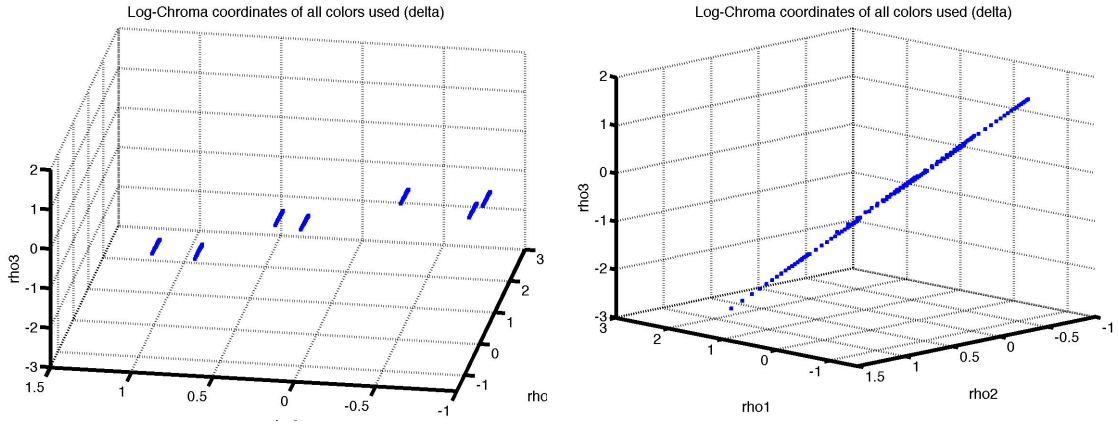
where s_w, s_{gm}, e_w and e_{gm} are replacement variables to simplify the equation. Equation (2.12) is a straight line parameterized by T sometimes referred to as Log-Chromaticity Difference (LCD). Notice that the two vector directions $(e_w - e_{gm})$ is independent of the surface reflectance ($s \equiv ln(s_w/s_{gm})$), although the line for a particular surface has an offset that depends on s_w. Every such line is parallel, with slope dictated by $(e_w - e_{gm})$. An invariant image can be formed by projecting these 3D logarithms of band-ratio chromaticity ρ (vector made of all ρ_w for a sample) into the direction $e^{\perp}$ orthogonal to the vector $e \equiv (e_w - e_{gm})$. The result of this projection is a single scalar which can be coded as a greyscale value.

To discover the appropriate invariant, suppose an image of particular surface reflectances $S_j(\lambda)$ under a set of m representative illuminants: $E_1(\lambda), E_2(\lambda), \ldots, E_m(\lambda)$. Assuming that the camera behaves approximately like a Dirac delta camera, the ρ_w should then all be approximately collinear. By subtracting the mean LCD, we can move those lines (one line per surface reflectance) so that they pass through the origin:

$$\mu^{1\ldots n} = \frac{1}{m}\sum_{k=1}^{m} \rho_k^{1\ldots n} \tag{2.13}$$

$$\Gamma_{1\ldots m}^{1\ldots n} = \rho_{1\ldots m}^{1\ldots n} - \mu^{1\ldots n} \tag{2.14}$$

where n is the number of considered surfaces, $\mu^{1\ldots n}$ is the mean of each considered surface, and Γ is the ρ demeaned. All Γ are used to form a $w' \times nm$ matrix M. w' is $w-1$ when a band is chosen for the ratio (if green (G) is used, then there are only two LCD: R/G and B/G) or w when the geometric mean is used (presented case). Fin (2001) note that, in Log-Chromaticity (LC) space, ρ is orthogonal to $u = 1/\sqrt{3}(1,1,1)'$. That is, ρ lives on a plane orthogonal to u (see Figure 2.6). A visual demonstration can easily be performed in Matlab (see Appendix B), it is less simple in a document. Two significant points of view are shown in Figure 2.6 to help make this point. The



(a) A view of the result of Equation 2.12

(b) Another view of the result of Equation 2.12

Figure 2.6: Two different views to show the plane in which the data belongs.

first view (Figure 2.6(a)) shows that all the lines are parallel and the second view (Figure 2.6(b)) the plane orthogonal to u.

The coordinate axis along which the variation (or variance) that is due to illumination is minimized needs to be identified. Fin (2001) note the covariance matrix $\Sigma(M)$ can be uniquely decomposed as,

$$\Sigma(M) = \frac{1}{nm} M M^t = U^t D U \tag{2.15}$$

where U is a rotation matrix and D is a strictly positive diagonal matrix whose diagonal terms are in decreasing order. The diagonal entries of D are the variances of M under rotation U. Basically, U and D can be found by calculating the Single Value Decomposition (SVD) of Γ. U can be used to represent the three vectors ρ into a two vector coordinate system in the plane

$$\chi \equiv U\rho. \tag{2.16}$$

As is shown in the Figure 2.7, χ_1 is in the direction of changes in illumination and χ_2 is in the direction Fin (2001) [13] associates to change in color. The 1D greyscale image is formed using only χ_2. Fin (2001) conclude that they can form an illuminant invariant image as long as the camera response follows the diagonal model. Worthey *et al.* [42] found that so long as a camera

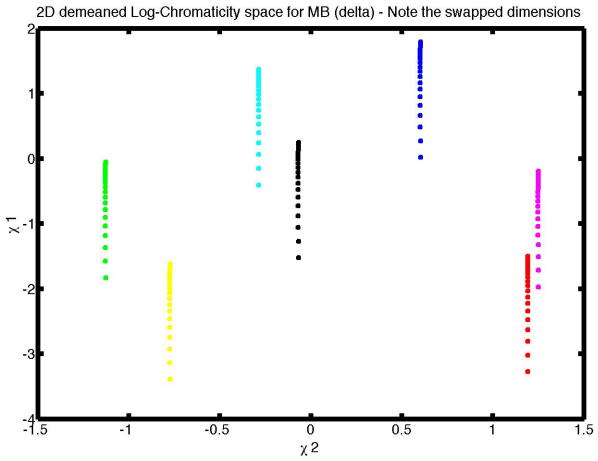


Figure 2.7: Result of Equation 2.16 - Note χ_1 and χ_2 have been swapped.

is equipped with fairly narrow sensitivities, e.g., with a support of 100 nm, the diagonal model will hold. However, in a series of works, Finlayson, Drew, and others (references 28,29,40 and 41 from [13]) found that new narrower-band sensitivities could be formed from broadband sensitivities by calculating an appropriate sharpening transform.

Starting in 2004, Fin (2004) [11] transition from using the 1D LC space (a shadow free space at the cost of color information) to using the 2D LC space. Essentially, χ_1 and χ_2 are used instead of projecting all the data onto χ_2. The 2D LC space is a chromaticity space but only χ_2 is illuminant invariant.

Training data (color samples under a number of illuminants) was necessary to obtain the invariant transform for a camera. Fin (2004) [11] use a different approach for an image from an unknown camera. The LCD for each pixel (Equation 2.12) is calculated. Because there are no sample set from the surfaces in the image, the data cannot be demeaned, but all ρ are in the same plane. Calculating the SVD allows the projection of the three vectors of ρ into the χ plane (as per Equation 2.15 and 2.16). The cloud of data is now in 2D and oriented as to maximize the variance in the first dimension. In the case of an image data set, this is not necessarily the invariant direction. The correct angle θ to rotate the data in the plane is such that the entropy for the marginal distribution along a 1D projection line orthogonal to the lighting direction is minimized. The greyscale image Υ along this line is formed via

$$\Upsilon = \chi_1 cos(\theta) + \chi_2 sin(\theta) \tag{2.17}$$

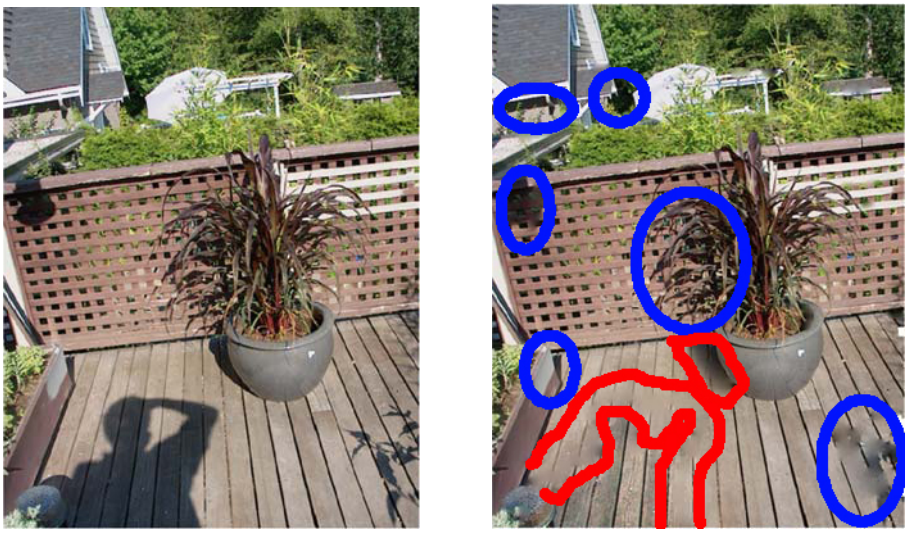
and the entropy is given by

$$\eta = -\sum_i p_i(\Upsilon) ln(p_i(\Upsilon)) \tag{2.18}$$

where $p_i(\Upsilon)$ is the probability of a value in Υ. The approach for this case, is to calculate the projection Υ of the 2D log-chromaticity representation of the image for $\theta = 0 \ldots 180°$ and calculating η for each θ. The minimum entropy is the correct projection for illumination invariance.

Fin (2006) [14] uses the LC space in a shadow removal algorithm of their own. The method consists of creating an edge map from the original image and from the 2D LC image. Edges appearing on both edge map are considered changes in surface reflectance, while those that only appear in the original image are considered changes in illumination. They use values from non-shadow pixels to re-integrate intensity in the image. This method functions for Planckian sources (day time images) with strong shadows (hard edges) as is demonstrated in Figure (2.8).

Both images from Figure 2.8 are from one of Fin latest paper, but the blue and red regions were added using Photoshop to highlight the regions discussed now. The blue ellipses show regions where shadows were not removed or mishandled. All the overhangs in the image kept their shadows, this can be observed at the edge of the roof and under the top railing. Although it is not mentioned in the paper, perhaps the algorithm was only run on the deck alone instead of the whole image. The top left of the railing is probably very challenging for their technique considering the high number of edges (grid pattern) in and out of shadows in a small region. The shadow from the plant on the grid pattern are also unchanged. The last obvious unmodified shadow surrounds the garden box on the bottom left of the image. The modified shadows are confined to the deck and the effect of the algorithm is observable.The shadow of the plant pot is partially removed (red circle at the center), it seems blurred in favor of the brighter deck pixels. Some of the shadows from the of image leaves (bottom right) have been removed, but some are still visible. The red region at the bottom traces the blurred region caused by the silhouette of a person. Only the edge of the shadow is blurred, note the center of the silhouette is not blurred.



(a) Original Image (b) Image with some shadows removed

Figure 2.8: An example of Finlayson and Drew shadow removal method [12]

2.6 Bilateral Filter

The original bilateral filter was presented by C.Tomasi and R.Manduchi at the IEEE International Conference on Computer Vision in 1998 [5]. The bilateral filter blurs (smooth) images while preserving edges, by means of nonlinear combination of nearby image values. The bilateral filter produces no phantom colors along edges. The method is noniterative, local and simple. It considers geometric closeness and photometric similarity to obtain the new pixel value.

One of the possible objectives to using the bilateral filter is to remove noise from an image. The intuition is that images typically vary slowly over space, so near pixels are likely to have similar values. Noise causes this intuition to be wrong because it induces small random variations in all pixels; so all nearby pixels are usually slightly different. The noise values that corrupt these nearby pixels are mutually less correlated than the signal values. So by averaging nearby pixels, the noise is averaged away while the signal is preserved. The assumption of slow spatial variations fails at edges (due to change of material or illumination), which are consequently blurred by low-pass filtering. Hence the purpose of the research to design a way to low-pass filter an image while preserving edges.

Although Tomasi and Manduchi's research was focused on black and white images and three bands images (RGB, hue-saturation-intensity or CIE-Lab), the bilateral filter can be used on images with any number of bands. To prevent the filter from creating artifacts, like phantom edges, all bands must be used in the filter. Separate smoothing perturbs the balance of colors and unexpected color combinations appear. Bilateral filters, on the other hand, can operate on all bands at once and determine which pixels are similar and which are not. Only similar pixels are averaged together, and artifacts mentioned above disappear.

The original bilateral filter [5] has two functional parts. (1) The closeness filter: a 2D spatial low-pass filter (Gaussian in the presented case),

$$c(\xi, x) = e^{-\frac{1}{2}\left(\frac{d(\xi,x)}{\sigma_c}\right)^2} \tag{2.19}$$

where ξ are the spatial coordinates of the neighbouring pixels within the filter's kernel, x are the spatial coordinates of the pixel of interest, σ_c is the standard deviation for the closeness filter $c()$ [the spatial Gaussian] and $d()$ is the Euclidian distance two data points (α&β)

$$d(\alpha, \beta) = || \alpha - \beta || . \tag{2.20}$$

(2) The similarity filter: also a low-pass filter (also Gaussian) but it is applied to the euclidian distance of the pixels value (instead of spatial coordinate):

$$s(\xi, x) = e^{-\frac{1}{2}\left(\frac{d(f(\xi),f(x))}{\sigma_s}\right)^2} \tag{2.21}$$

where $f()$ is the value of the pixel and σ_s is the standard deviation for the similarity filter $s()$.

Depending on the flexibility desired, both filters can use a single standard deviation or a different one for each dimension. There are very few reasons to apply this in the closeness filter (giving more importance to a spatial direction), but the similarity filter can benefit from it. Especially if you are not using a RGB space. For instance using a HSL space, a smaller standard deviation for the hue, a bit bigger for saturation and rather large for intensity would be preferable (accounting for the domain of each dimension and by the changes of least perceivable difference). In either case, by using the filter theorem, all the filters can be applied to the pixel of interest at once as

(a) Original Boy Image (b) Filtered Boy Image

Figure 2.9: Bilateral filter example from original paper [5].

$$g(x) = \frac{1}{k(x)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d\xi \tag{2.22}$$

where

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, x) s(f(\xi), f(x)) d\xi$$

is normalizing the final filter, it ensures the sum of all the weights of the kernel are equal to one. Notice the closeness filter does not change as the kernel moves across the image and can be calculated once; the similarity filter must be calculated for each pixel.

Figure 2.9 is an image from the original paper. It shows one of the possible outcomes when using the bilateral filter (the cartoony look). Tomasi and Manduchi have evidently designed an efficient and simple way of blurring an image while protecting high frequency information. All edges (either due to illumination or changes in material) are preserved without creating phantom color images (like a green contour to the red shirt). Even the variations in hair color are protected. This is one possible outcome. The authors concluded the paper by explaining that the application dictated the parameters to be used for the filter. It could also be said the content of the image filtered as a considerable effect on the parameters. The amount of noise present, the spatial distance between edges, the smoothness desired in the final image will impact the parameters to be used. Parameters too loose will make a bilateral filter acts as a regular low-pass filter. Parameters too strict and the filter will be able to average very few pixels.

2.7 Metric of Success

Since the goal of this research is to enable computer vision to do better in various illumination conditions, the intent is to demonstrate quantitatively the effect of the presented algorithm on a computer vision task. This section covers the applications selected to test the presented algorithm, measure its effect and the known method used as a benchmark.

2.7.1 Applications

ENVI (by ITT Visual Information Solutions) [19] is an imaging software suite available to the Center of Imaging Science at RIT which has integrated classifiers and post-classification tools. The functions to read/write images in the ENVI format were downloaded from the link below [2]. "Minimum Distance" is a simple and well documented supervised classifier. It is considered a supervised classifier because it requires training data. This training data can be specified via Regions of Interest (ROI). Those ROIs are an ensemble of pixels assumed to belong to a single class. The classifier assigns a class to a pixel based on the shortest Euclidean distance between the pixel's value and the mean value of the class [39]. *k-means* is also a simple, well documented classifier, and is unsupervised. It is an iterative algorithm; it starts with random class means, assigns every sample to a class the same way the Minimum-Distance classifier does, then recalculates the mean for each class and re-assigns the samples using the new mean. The process is repeated until the mean of each class doesn't change [39].

2.7.2 Statistics

A confusion matrix is a way to evaluate the performance of a classifier. It is a $k \times k$ matrix, k being the number of classes compared, where the truth classes are in columns and the rows contain the image-derived classes. Each cell contains the number of pixels from a given truth class associated to a given image class. A perfect classification is a diagonal matrix, all pixels assigned to the correct class. The confusion matrix tool in ENVI provides the matrix, the simple accuracy given by

$$p_o = \frac{\sum_{i=1}^{k} c_{ii}}{N} \tag{2.23}$$

[2] http://www.mathworks.com/matlabcentral/fileexchange/27172

where c_{ii} are the cells from the diagonal of the confusion matrix and N is the number of samples, the kappa coefficient ($\hat{k}$), the commission and omission error to name only those of interest in this research. The simple accuracy is a good indication of how many pixels were rightfully assigned, but it does not penalize a classifier that failed to classify pixels or that assigns pixels to a wrong class. The $\hat{k}$ is based on the simple accuracy but it removes the portion of the accuracy due to random chance and is calculated as follows:

$$p_c = \frac{1}{N^2} \sum_{i=1}^{k} c_{it} c_{ti},$$

$$c_{it} = \sum_{j=1}^{k} c_{ij},$$

$$c_{ti} = \sum_{j=1}^{k} c_{ji},$$

$$\hat{k} = \frac{p_o - p_c}{1 - p_c} \tag{2.24}$$

where p_c is the probability of correct classification due to random chance, c_{it} is the sum of each row and c_{ti} is the sum of each column.

The commission error is the ratio of pixels wrongfully assigned to a class over the number of pixels actually in the class (truth). The omission error is the ratio of pixels from a class not assigned to that class over the total number of pixels in the truth class. Additional information on classifier evaluation can also be found in [39] or ENVI's help file [19].

2.7.3 Benchmark for Improvement

For comparison, a common method to eliminate the intensity in an image (reduce the impact of changes in illumination) was also included in the experiments. To distinguish it from the normalization by the geometric mean, it is referred to as the Spherical Unit Normalization (SUN) and it is expressed by the following equation:

$$p'(x, b) = \frac{p(x, b)}{\sum_{i=1}^{B} p(x, i)} \tag{2.25}$$

where p' is the new pixel value, p is the input pixel value, x is the spatial coordinate of the pixel, b is a band of a B bands image.

2.8 Summary

The literature review revealed that the problem was known and well understood, but that there is presently no ideal solution. Ideally to achieve an automated persistent surveillance system, objects would have to be identifiable in changing illumination conditions (day and night) in unknown environments. There are two principal areas of research concerned with this problem: shadow removal (see Section 2.1) and color constancy (see Section 2.4) algorithms. Either solution is too limited in scope. Shadow removal designates algorithms embedded in segmentation to prevent shadows to be included in the foreground. They require backgrounds built from video sequences (known environment) and functions for a specific illumination condition. Most color constancy algorithms are designed for a single source and they assume that all pixels in an image are under that illuminant. They would allow the recognition of an object found in two images each with a different illuminant, but they are not normally applied to the retrieval of data from shadows. Fin (2006) [14] is the single instance of an algorithm to correct pixels in shadows to the main illumination. That algorithm is limited to nearly Planckian sources (day time outdoor) and shadows with hard edges.

The research proposed in this thesis focuses on a solution detached from any specific computer vision application (unlike shadow removal tied to segmentation). It would be applied to a single frame before submitting the image to an application. The proposed solution will require as few assumptions about the environment and illumination as possible to be usable in the widest range of situations (not limited to specific sources and/or shadows).

Chapter 3

Implementation

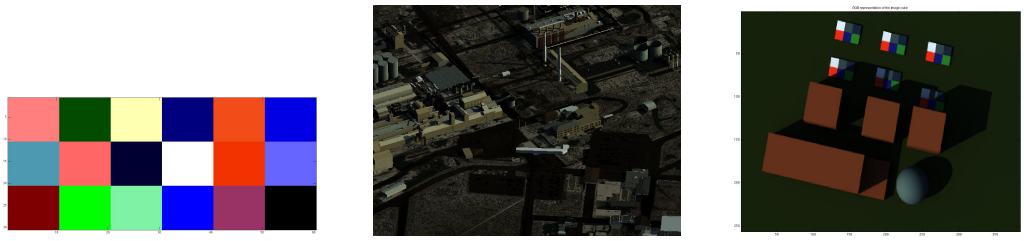
This chapter covers the theory and the practical details of the methods developed in this research. It begins with the origin of the images used to test the proposed algorithm. It then goes over the experiments on the Log-Chromaticity (LC) space (Fin (2001)) that leads to the novel color space using hyperspectral data. The next section describes the modification made to the bilateral filter, followed by how the Hyper-Log-Chromaticity (HLC) and Illumination Equalization Filter (IEF) were combined.

3.1 Data Used

The data used to develop the proposed algorithm is entirely synthetic. This section discusses the scenes that were used, the illumination sources that were part of the research and included in the scenes and the various combinations of scene and sources that were used throughout this research.

3.1.1 Scenes

Three scenes (Figure 3.1) were used throughout this research to verify and test the algorithms. The first scene used is the 3×6 checker board shown in Figure 3.1(a). It is the most simple scene (only two dimensional) with which Matlab functions were tested. The checker pattern is made in Matlab, contains a few pixels and few panels of colors meant to observe the effect of functions in the absence of cosine illumination, background adjacency and shape factor to the sky effects. The



(a) Synthetic. (b) Portion of Megascene2. (c) Simple Simulated.

Figure 3.1: The three scenes used in this research.

structure (fixed grid) of the scene and its size (30 by 60 pixel) made it easy to verify each step of a tested function. The color of each panel was set manually by changing RGB values.

The second scene (Figure 3.1(c)) was the most used to verify different properties of the algorithm. Blender [18] was used to create a simple scene containing eight materials. The six 2×3 panels are exact copies (red painted wood, blue painted panel, green painted panel, green roofing shingle, asphalt and blue car paint), the sphere is made of the same material as the asphalt panels. All surfaces of the blocks are of one material (red brick), the grass is also a single material. All the materials used are recorded spectra from a database. The blocks allowed shadows to be cast on the panels and the sphere was a good form to observe light falloff.

The scenes were rendered using the Digital Imaging and Remote Sensing Image Generation (DIRSIG). *"The DIRSIG model is a first principles based synthetic image generation model developed by the Digital Imaging and Remote Sensing Laboratory at Rochester Institute of Technology. The model can produce multi- or hyper-spectral imagery from the visible through the thermal infrared region of the electromagnetic spectrum. The model can be used to test image system designs, to create test imagery for evaluating image exploitation algorithms and for creating data for training image analysts.*[1]*"* [26]

The last scene used was a portion from Megascene2 (Figure 3.1(b)). It is a very big 3D scene representing an industrial plant in a desert environment. The lack of color and the number of pixels makes it a more complex scene for the algorithms. It contains six different materials and the images were also rendered using the DIRSIG.

[1] http://dirsig.cis.rit.edu/

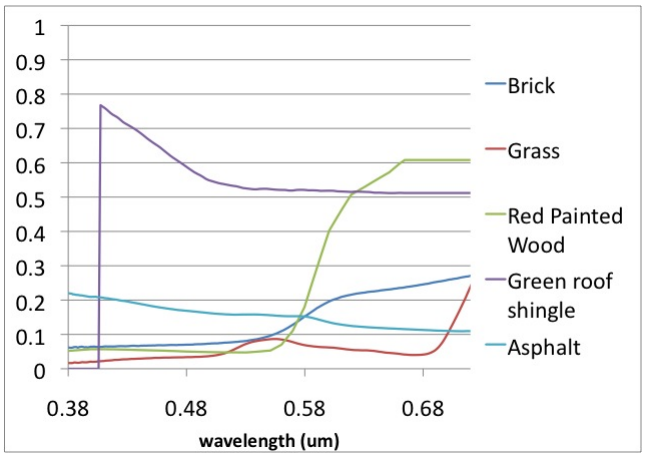


Figure 3.2: The spectra used for materials in the simple scene (Figure 3.1(c)).

Material Spectra

Figure 3.1.1 shows the spectra of some of the materials used in the simple scene. Note that these were not used to train the algorithm.

3.1.2 Sources

This research started mainly using the Planckian sources to which the HLC color space was attuned. The blackbody radiators were all simulated using Planck's equation (Equation 2.8) in Matlab as can be seen in functions available in the appendices. As research progressed, the spectra of the solar, lunar and sky propagated through the atmosphere using the MODerate resolution atmospheric TRANsmission (MODTRAN) model were introduced. MODTRAN is a computer program designed to model atmospheric propagation of electromagnetic energy [1]. These are more realistic sources instead of theoretically perfect sources. Three non-Planckian sources that had been included in the DIRSIG database were also used. Those sources are spectra that were measured around the campus of RIT. One is a high-pressure sodium lamp, the other is a mercury vapour lamp and the last one is a bare fluorescent source. The spectra of all the sources mentioned above are shown in Figure 3.16. The last set of sources (see Table 3.1) to be included in this research were obtained from the NOAA website and includes a wide variety of fluorescent, sodium and mercury vapour lamps [21].

Table 3.1: Sources from the NOAA file used in this research.

High Pressure Sodium Lamp	70W
	150W
	1000W
Mercury Vapor Lamp	Iwasaki 60W
Fluorescent Lamp	OCTRON 32W
	low CRI
	High CRI
	5000 K
	4100 K
	3500 K
	3000 K
	CL Nvision 9W
	CFL Greenlite 13W

3.1.3 Captures

The color checker scene was used for all of the initial testing. It was used without any further modification to test the Illumination Equalization Filter (IEF) as shown in Section 3.4. It was then modified to be composed of panels of 10 by 15 pixels so it could be illuminated with three different sources. Each source illuminated a column of five pixels so that each column of panel was lit by all three sources. This way the effect of the HLC space could be observed for each panel as well as the behaviour of the algorithm. Three variations of the illuminated checker scene were created: the first one used the Planckian equation to simulate the spectrum of a moonlit sky ($\approx 4100K$), the sun ($\approx 5500K$) and daylight ($\approx 6500K$) (see Figure 3.1.3), the second version used the three spectra propagated through MODTRAN mentioned above (see Figure 3.4), and the last one used the sources found on the RIT campus (see Figure 3.5). The training set for those images used bands centered on 450, 550 and 650 nm (Gaussian responses - RGB). The set makes a good visual example of the difficulty of identifying a color with changing illumination.

The scene created with Blender [18] has five variations. The first version is a simulation of a day scene with the sun at 42° zenith, the solar radiance propagated through the atmosphere and the sky light as per MODTRAN (day). The next version is an example of a moonlit scene

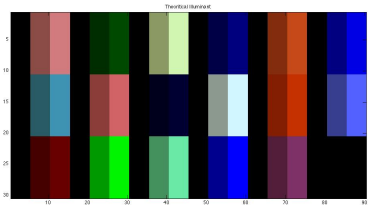


Figure 3.3: The checker scene (Figure 3.1(a)) illuminated by the Planckian equivalent of the night sky, the day sky and daylight (solar and sky).

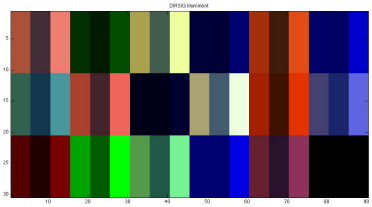


Figure 3.4: The checker scene (Figure 3.1(a)) illuminated by day sky, solar and daylight as per the MODTRAN model.

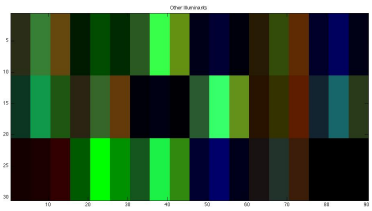


Figure 3.5: The checker scene (Figure 3.1(a)) illuminated by a bare fluorescent, a high pressure sodium lamp and a mercury vapor lamp.

(moon). Then there is a version with a tungsten lamp at the top of the right ledge of the bigger block (tungsten). The other one has a high-pressure sodium lamp instead of the tungsten lamp (sodium). The last version has a tungsten lamp at the top of the far left block and a high-pressure sodium lamp at the top of the far right block (moon-tungsten-sodium (mts)). Those images use wavelengths between 380 and 720 nm with increments of 20 nm (18 bands).

The portion of Megascene2 has two versions. One is a day version with a solar and sky irradiance calculated by MODTRAN (MSday). The other version was a night scene with implanted sources (the ones found on the RIT campus) (MSnight). Those images use wavelengths between 380 and 720 nm with increments of 2 nm. Because those data cubes were too large to be handled on a personal computer, the presented algorithm was run on a subset of bands, the same as the simple scene. In addition to large memory requirements, the product of 171 irradiances for some pixels was beyond the limit of double precision data types (giving the result positive infinity).

3.2 Hardware & Software

The synthetic data, the HLC and the IEF were created and tested on a laptop. It was a MacBook Pro 6,2 with an Intel Core i7 (2 cores) and 8 GB of RAM running Mac OS X 10.6.1. The functions were written using Matlab R2009a for Mac. The imaging applications were from ENVI 4.6.1 for Mac OS X 64. The runtime of the proposed algorithm (merging of HLC and IEF) will vary with the number of pixels and the parameters used. On average, each pass on the simple scene took 15 seconds using the system described above. Filtering the portion of Megascene2 took almost fives minutes a pass. No efforts were made to make the algorithm run faster.

3.3 Color Space

This section covers the origins of the HLC space, as well as how to implement and use it.

3.3.1 Reproduction of Log-Chromaticity

In order to get a proper understanding of Finlayson *et al.* (2001,2004,2006,2009) work on the Log-Chromaticity (LC) space, the starting point was reproducing the figures from their 2001

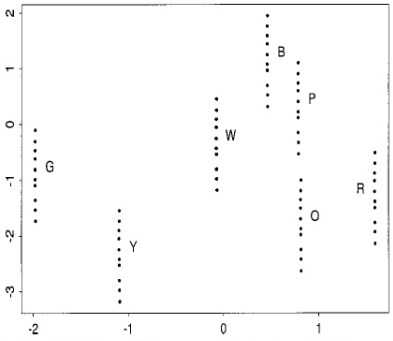


Fig. 7. Perfect Dirac delta camera data (sensitivities anchored at 450, 540, and 610 nm). LCD's for seven surfaces (green, yellow, white, blue, purple, orange, and red) under ten Planckian lights (with increasing temperature from 2800 to 10,000 K). LCD's (from Fig. 6) have been rotated so that the x coordinate depends only on surface reflectance; the y coordinate depends strongly on illumination.

(a) fig. 7 from [13]

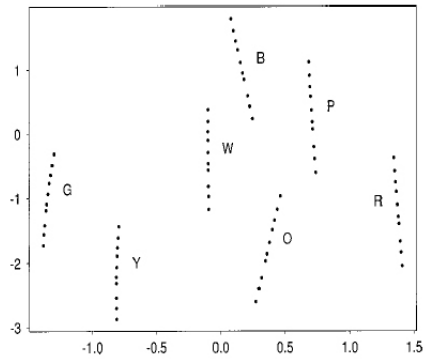


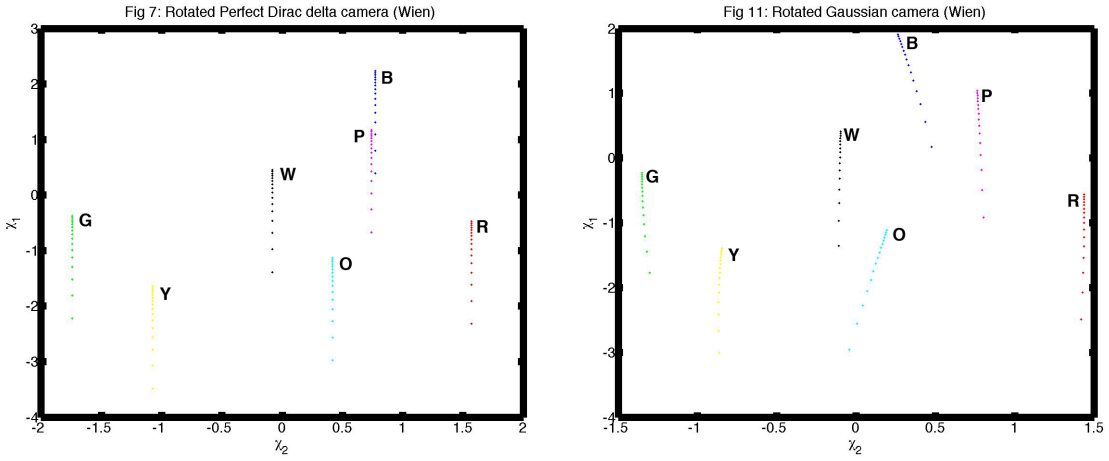
Fig. 11. SONY DXC-930 camera data. LCD's for seven surfaces (green, yellow, white, blue, purple, orange, and red) under ten Planckian lights (with increasing temperature from 2800 to 10,000 K). LCD's (from Fig. 9) have been rotated so that the x coordinate depends only on surface reflectance; the y coordinate depends strongly on illumination.

(b) fig. 11 from [13]

Figure 3.6: Figures 7 & 11 from the original paper [13].

paper [13]. The Matlab code to reproduce these figures is presented in Appendix A. The figures of main interest are the reproductions of Figure 3.6 in the original paper that show the result of the illumination invariant camera calibration. Note that the following color patches from the MacBeth color chart [17] are used: Red(R), Green(G), Blue(B), Yellow(Y), Orange(O), Purple(P) and White(W).

Overall the reproductions are very close to the original versions, enough to be convinced the LC transform could be used. A few differences can be noticed. When looking at the figures resulting from a camera with Dirac delta camera response, the purple samples from the presented research falls directly under the blue. The orange data in Fin (2001) is under the purple data while in the presented research it is between the white and blue data. Those are not essential differences and no attempt to correct them were made since the code reproduces the results for the other colors perfectly. Note that a similar but different simulated Gaussian camera response was used. Fin (2001) used a smaller blue response than the green and red responses. The responses in the presented research are centered the same all share the same transmission.



(a) (fig. 7) Dirac delta camera response.

(b) (fig. 11) Gaussian camera response.

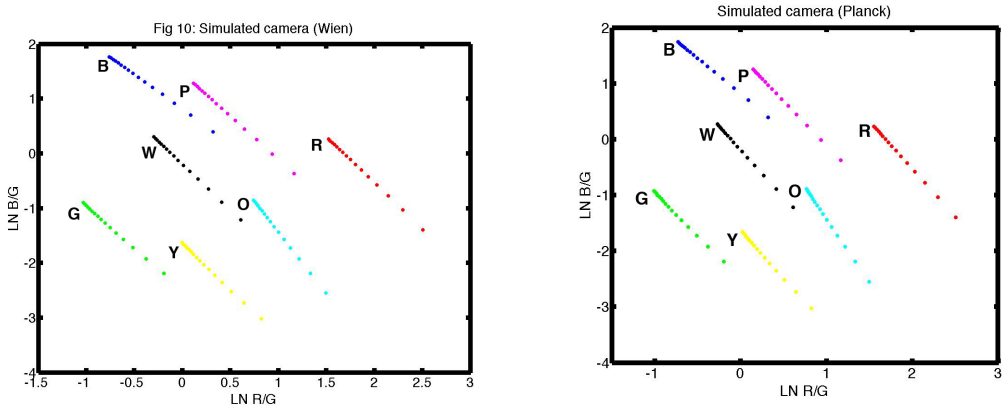
Figure 3.7: Reproductions of Figures 7 & 11 from the original paper [13].

Wien vs Planck

The impact of using Wien's approximation (Equation 2.9) over Planck's equation (Equation 2.8) was studied by using both type of sources in parallel throughout the reproduction (see Figure 3.8). All the figures are exactly the same whether using Wien or Planck's equation. From that point, even though the mathematical proof is done using Wien's approximation, Planck's equation is used to simulate blackbody radiators.

Colors in LC Space

In an effort to better understand the 2D LC space (from this point on only the 2D version is considered since the color information is required), additional experiments were conducted that can also be found in my archived functions. Instead of using actual color patch samples, different colors were synthesized by changing the transmission of each channel (i.e. to create a blue sample, the transmission for the blue channel is set to one while the transmission for the green and red is set to zero). Four fixed colour patches: pure blue, pure green, pure red and pure white ("Pure" refers to the fact that no other color was present) served as anchors in the colour space to study the position of the fifth patch which was given a multitude of different values. A similar range (to the previous experiment) of simulated blackbody (BB) radiators (Planck) were used as sources. This



(a) Reproduction of fig. 10 from Finlayson's 2001 paper [13].

(b) Same experiment with sources simulated using Planck's equation.

Figure 3.8: Comparison between Wien and Planck.

experiment exposed some behaviour of the LC space and permitted a number of vital observations to be made:

1. Whatever the transmission used, if it is equal for all three channels, the color matched the location for the pure white.

2. A color will always be found closest to the color anchor corresponding to its highest transmission.

3. If two channels are kept even and the third one is varied; the location moves along a straight line passing though the white anchor and the color anchor corresponding to the varied transmission.

4. If two channels are kept constant (but not even) and the third is varied from one to zero; the location moves along a straight line passing through the white anchor and the starting location.

5. When trying to understand color in the Log-Chromaticity space, an observer should pay closer attention to colors under a single illuminant. This is made clear by Figure 3.9.

When the focus is on all colors under a specific illuminant, the result is a chromaticity space much like a^*b^* (from CIE-Lab), XY (from XYZ) or the hue-saturation space from any hue-

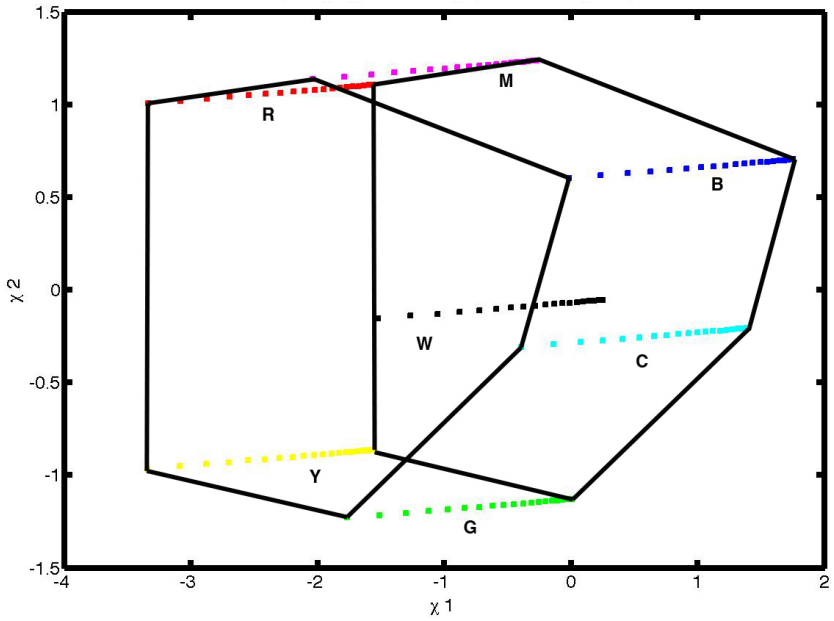


Figure 3.9: Color space in Log-Chromaticity space (Planckian sources).

saturation-intensity color space. The intensity has been properly removed by the transform, so it is not present in this space. The order of the color patches in their experimentation makes sense when you go around the white patch in a circle. The red, green and blue samples are roughly 120° apart, and the other colors where you would expect them in a hue-saturation space (type of chromaticity space). The yellow and orange between the green and red, the purple between the blue and red. Basically, with the intensity removed, the Log-Chromaticity space for a single illuminant is a slice of the HSL space represented in Figure 2.4. The neutral colors are found at the centre (near the white patch in LC) and the more colorful samples are found away from the white patch.

This means the Log-Chromaticity space shares the liability of any other hue-saturation space with neutral color. It also means that a single dimension cannot be sufficient to identify a color, definitely not with cartesian coordinates. It might be manageable with a polar coordinate system (using only hue), but there would still be an issue with neutral colors. There is an important difference between the LC space and other known chromaticity spaces. Unlike CIE-Lab, XYZ or HSL which are bound to fix domains (the human visual response for CIELab and XYZ and the

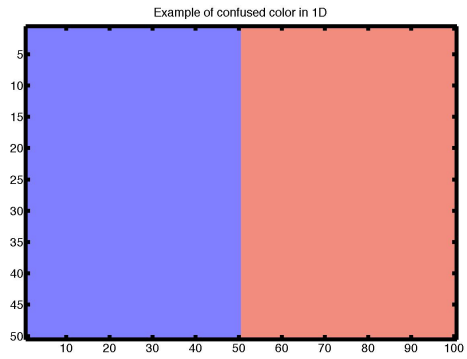


Figure 3.10: An example of confused color, if only one dimension is used. Both colors RGB coordinates in Matlab: left [0.5,0.5,1] and right [0.95,0.55,0.5].

image minimum and maximum for HSL), the LC space is limited by the camera's response and the reflectance of materials. This experiment confirms s (surface reflectance in Equation 2.12) requires both dimensions of the LC space (confined to a plane which by definition only has two dimensions), otherwise colors could be confused (see Figure 3.10 an example of confusion).

Illuminants (Planckian) in LC Space

Now that the behaviour of the Log-Chromaticity space with one illuminant is understood, the whole set as presented by Finlayson et al. was studied. The space occupied by each illuminant (polygons of Figure 3.9 and 3.11) is exactly the same except for a translation in a given direction (vector e). This is consistent with the desired outcome of the process. The color samples of two different illuminants are joined to demonstrate this on Figure 3.9. This polygon fits all illuminants of any temperature within the set domain. Since the translation is in the same direction for all Planckian illuminants, this is consistent with Fin (2001) conclusion that e is unidimensional and one dimension of the LC space is illuminant invariant (since s requires both dimensions).

Other Illuminants in LC Space

Given the interest for a solution for day and night environments, the second environment implements sources far from the Planckian assumption. Other types of sources were introduced in the training set. At first the spectrum from three sources previously recorded around the RIT campus

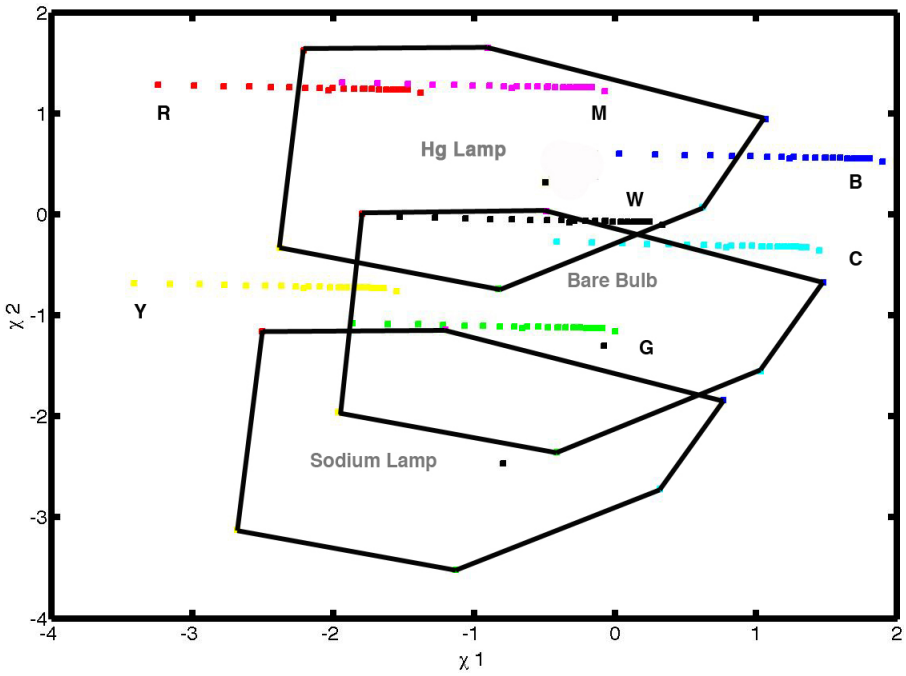


Figure 3.11: Color space in Log-Chromaticity space (all sources types).

for another thesis and saved in the DIRSIG database (a bare fluorescent, mercury vapor lamp and a high-pressure sodium lamp) were used. Later on, the spectra from a study carried out by the National Oceanic and Atmospheric Administration (NOAA) [21] on various sources were also used to study the LC space. In essence, the rotation eliminates the requirement of a second dimension to describe the vector e when the illuminant is Planckian. This observation is limited to Planckian illuminants, because as it is demonstrated in Figure 3.11, the polygon defining the color space of an illuminant is valid for non blackbody radiators, but the translation direction is not necessarily the same as Planckian illuminants. This is the experiment that introduced the concept of calibration of the LC space. The LC space is more useful with e unidimensional, which can be achieved for (at least) any two sources if the space is rotated accordingly. Otherwise, both dimensions of the LC space contain illuminant and surface reflectance information. This also means that for the purpose of this research, the Equation 2.12 must take a more generic form and drop the dependency to a blackbody temperature (since non-Planckian sources don't have one). The new formulation is written as

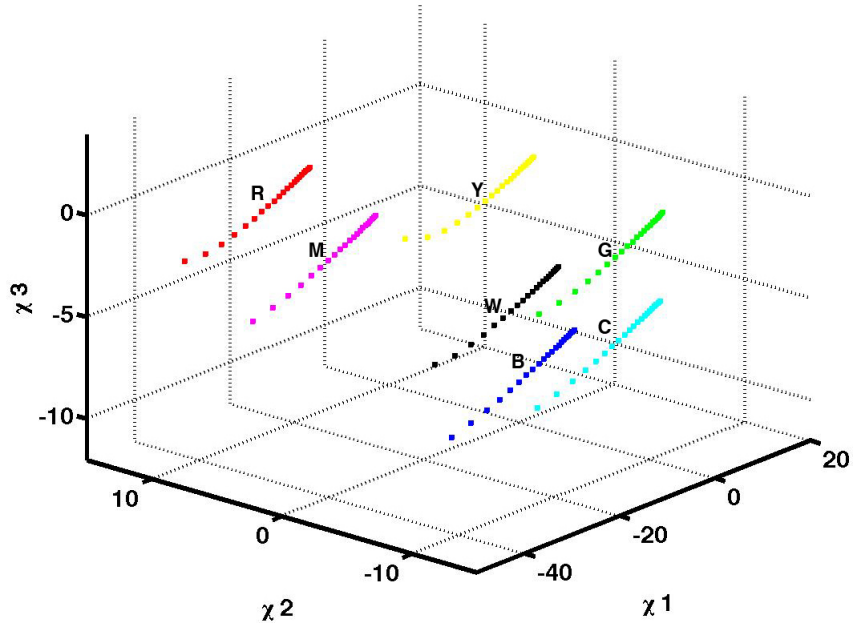


Figure 3.12: The top three dimensions of Equation 2.16 for a hyperspectral training set.

$$\rho'_w = s + e. \tag{3.1}$$

where ρ'_w is the new expression of ρ_w (Equation 2.12).

3.3.2 Hyper-Log-Chromaticity Space

In their 2006 paper [14], Finlayson and Drew mention the study of a four band camera allowing to remove specularities. All things considered, there is no theoretical limit to the number of dimensions for the training set. The SVD in the process will provide an estimation of the information content (assuming variance is correlated to information) of each dimension. Theoretically, the Fin (2001) algorithm should still separate surface reflectance and illumination color. This was also a relevant experiment as the Center of Imaging Science (CIS) at RIT has access to means to simulate or capture hyperspectral data. The focus was initially on the top three dimensions.

Figure 3.12 shows the top three dimensions from Equation 2.16 on an hyperspectral (380-720nm sampled every 1 nm by a Dirac delta function) training set of seven color patches from the MacBeth

color chart [17] (Red(R), Green(G), Blue(B), Cyan(C), Yellow(Y), Magenta(M) and White(W)) with blackbody radiators (2000K to 7000K in 250K increments). Since this space is built on the same theory as the Log-Chromaticity space, but requires hyperspectral data, we refer to it as the Hyper-Log-Chromaticity (HLC) space. The linear relation between colors under different Planckian illuminants is still present, but the data is no longer confined to a plane anymore. In LC space, each illuminant occupied a portion of the plane, by default all illuminants in LC space occupy a parallel plane. The parallelism of illuminants seems to be maintained in HLC (when considering Planckian sources), but each illuminant occupies a surface instead of a plane.

The existence of this third dimension should allow us to take a step closer to achieving illuminance invariance. The LC space has two vectors (s and e) of two dimensions confined in a plane (two dimensions available). At best it can isolate the illumination to a single dimension, making one dimension of the surface reflectance illuminant invariant (not sufficient to describe color). Theoretically, the HLC space has the same vectors in a volume. It could isolate the illumination to a single dimension and make the surface reflectance entirely illuminant invariant. Finlayson's *et al.* original method to rotate the space will function in HLC space without requiring any modification (projection on eigenvectors of the demeaned data set). The Entropy Minimization (EM) method will require some changes to account for the additional dimension.

Figure 3.13 is the projection of the data onto χ_1 and χ_2 after the HLC space (Figure 3.12) has been reoriented to maximize the variance in the first dimension. The result is identical to the LC space.

Figure 3.14 is the projection onto χ_2 and χ_3 of the same data shown in Figure 3.12 and 3.13. The result is not perfectly illuminant invariant, but all color samples are more tightly clustered by color whatever the Planckian illuminant. This was to be expected since, as can be observed in Figure 3.13, the lines for each color are slightly curved. The LC space was limited to a short range of temperatures, but the HLC space achieves near illuminance invariance on the complete range of temperatures of interest. The similarity to a chromaticity space is even more apparent with the projection onto χ_2 and χ_3.

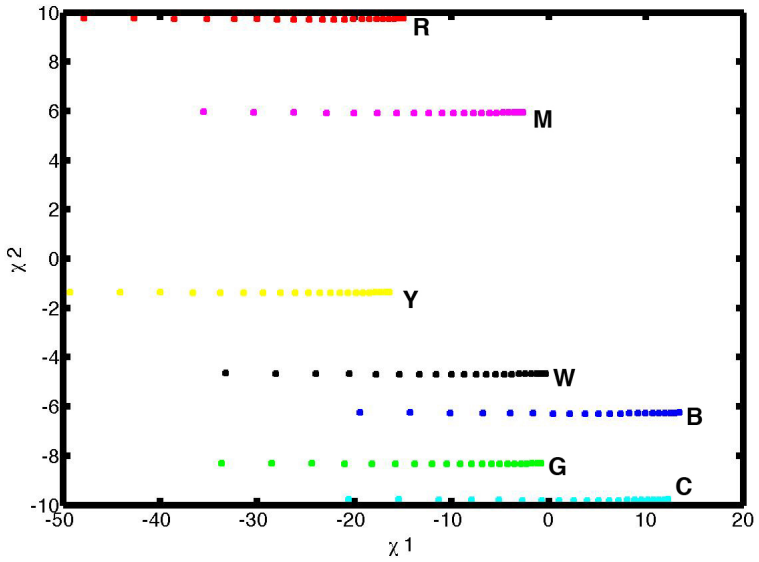


Figure 3.13: Finlayson's projection after the rotation of the data from Figure 3.12.

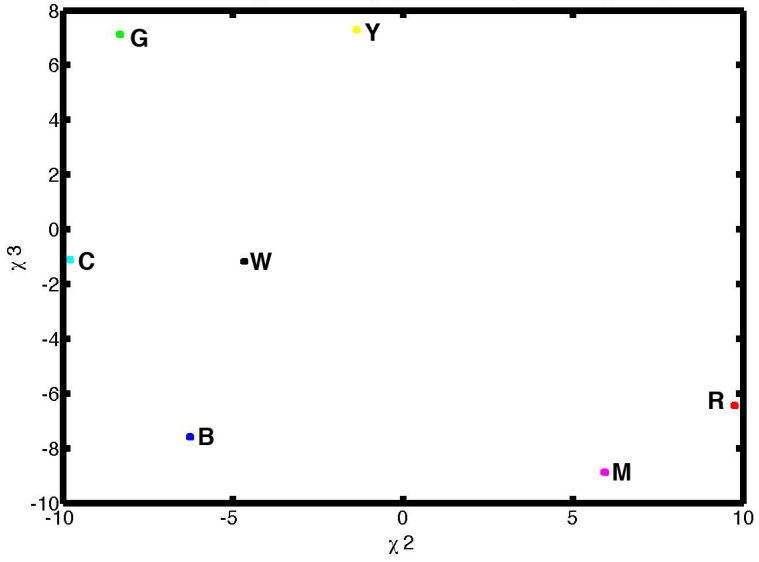


Figure 3.14: The new projection after the rotation of the data from Figure 3.12.

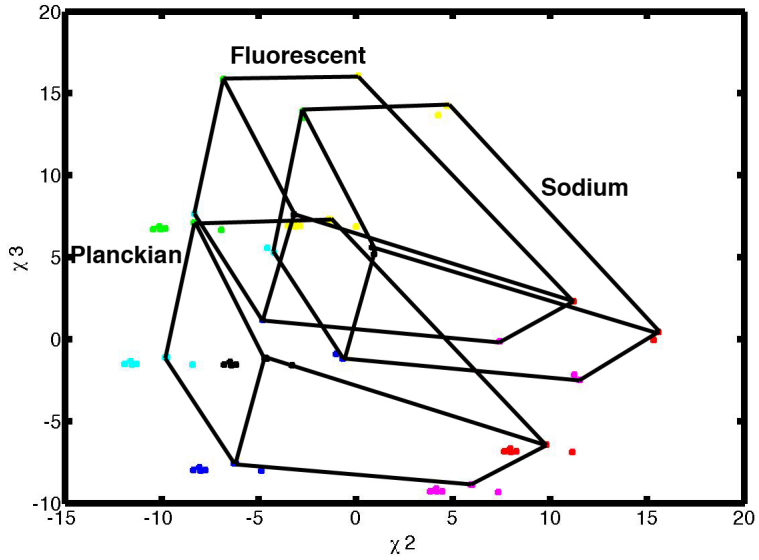
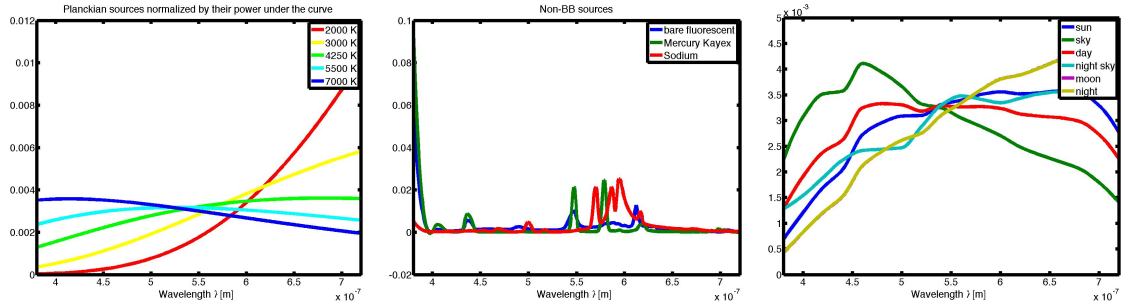


Figure 3.15: Same training and calibration as Figure (3.14) but with other types of illuminants included.

Other Illuminants in HLC Space

Similarly to the study of the LC space, non-Planckian sources were also included in the study of the behaviour of the HLC. Much like the LC space, all color patches have the same relative 'structure' for each illuminants. The non-Planckian sources seem to occupy surfaces roughly parallel to the surfaces from the blackbody radiators but translated in the volume of the HLC space. This was observed by rotating the volume manually in Matlab and verifying that various sources could be made to have corresponding projections. Although this property was not used in this research, the HLC could be calibrated to be illuminant invariant to other sources then near Planckian sources. Figure 3.15 shows where the non-Planckian sources are placed in the HLC space calibrated for blackbody radiators.

The polygon labeled 'Planckian' marks the location of the samples in the previous figure (Figure 3.14) while the polygon labeled 'Fluorescent' is for a bare fluorescent and 'Sodium' is for a high pressure sodium lamp. The cluster to the left of the planckian samples are the spectra modelled by MODTRAN for solar, solar+sky, lunar+sky and sky radiances. The sample immediately to the right of the Planckian samples is the lunar radiance modelled by MODTRAN and the samples



(a) Examples of spectra from Planckian sources.

(b) The spectra from non-Planckian sources used.

(c) The spectra modelled by MODTRAN.

Figure 3.16: Normalized spectra used in this research.

near the sodium samples are from a mercury vapor lamp. Attempts were made to find a relation between the other illuminants and the blackbody radiators. An experiment sought to discover a relation between the coordinates in the first dimension of the HLC and the Correlated Color Temperature of the non-Planckian sources but there was no correlation. Although the relation cannot be proven with hard evidence, the position of each sources subspace does not seem random. This relation does not relate to the CCT but it seems to have a link to the relative color of the source.

Figure 3.16 contains the normalized spectra of most sources used (not all blackbody temperatures used are included) for the study. Multiple observations can be made to reinforce this impression of a relation between source color and its location in HLC space. The Planckian theoretical sources have very smooth, well behaved spectra while the two figures with recorded sources are not so well behaved. Actually, the man-made sources (Figure 3.16(b)) consist mostly of spikes, strong emissions in narrow bands, and the modelled irradiance (Figure 3.16(c)) has features. For the sake of this explanation, the Planckian samples are considered the standard for comparison. So the white samples under Planckian illuminants are the 'true' white and the other samples divide the chromaticity space in color. Looking at the non-blackbody sources, they have spikes in the range of wavelengths associated with green to yellow. Correspondingly, their white samples are close to the green and yellow samples from the Planckian sources. Similarly with the modelled spectra, the sky and day have features in the blue and they are found in the general direction of the blue sample. The moon spectra is very strong in the red (basically a red mirror reflecting solar

(Planckian radiance) and it is found in the general direction of the red sample. Again, those are not conclusive comments, but there is a hint of a relation between the non-Planckian sources color and its position in the HLC space.

Although not based on solid science, this hint of a relation is interesting when considering the scenario of sources present for which the HLC was not calibrated. The HLC will not be illuminant invariant for those sources but the coordinates given will not be more erroneous then directly from the sensor reaching irradiance (difficulty to differentiate a 50% reflector illuminated by 100% source from the opposite combination). That is if only considering the second and third dimension, the first dimension could be used to differentiate the two sources. Potentially, with the help of the first dimension, a white panel under a yellow light could be identified amongst yellow panels under a white light.

3.3.3 Training and Calibration

Since all the images used in the experiments were simulated, the training data had to also be simulated. Planck's equation was used to create the visible spectra of a wide range of blackbody radiators and the recorded spectra of the seven color patches of interest (RGB, CYM & white from the MacBeth color chart [17]) to create a training set matching the image/camera of interest. The basis for the training set used in this research is a $(color\_sample * sources) \times wavelengths$ (wavelengths or bands) matrix. The training set (temperature and color samples) were the same for all images, but the transform had to be re-calculated so that the bands (camera response) matched. The Matlab code to carryout those steps can be found in the appendices (Appendix B).

It is important to replace values of zero with a very small value (relatively to the values of the training set) or to bias the whole set. Leaving zeros in the training set would make the geometric mean of these samples also zero. A zero in the denominator gives infinity and is of little use. The method described here is meant for a training set made with more sources then colors because of the nature of the calibration steps. Once the proper training set is created, all ρ are calculated for every sample with

$$\rho_w(x) = ln\left(\frac{f_w(x)}{\sqrt[W]{\prod_{i=1}^{W} f_i(x)}}\right) \tag{3.2}$$

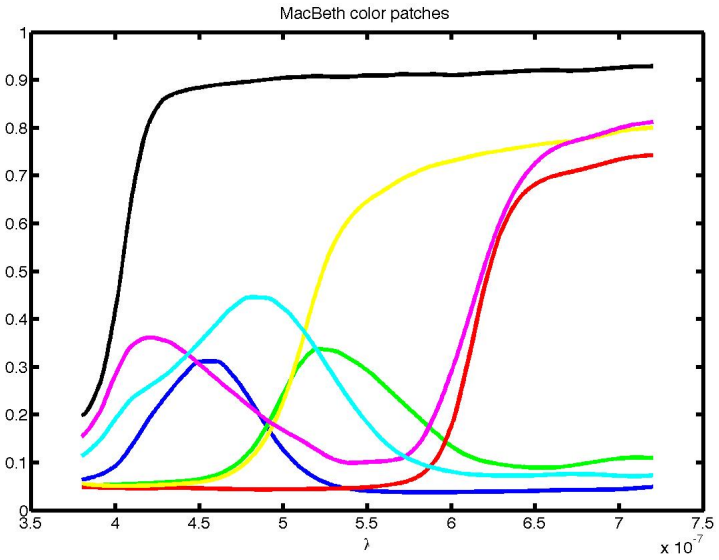


Figure 3.17: The normalized spectra of the MacBeth color patch used in this research.

where W is the number of bands in the image, $f_w(x)$ is the value of the pixel at location x for the band w.

The SVD for the training set is calculated and the input eigenvectors are recorded for later use with images from that camera. The transform is used on the training set as follows

$$\rho_w(x) = U * S * V'; \chi_w(x) = \rho_w(x) * V \tag{3.3}$$

where U is eigenvector "output" matrix, S is the diagonal eigenvalue matrix and V is the eigenvector "input" matrix (transform matrix) and only the top three bands $\chi_{1,2,3}$ are conserved. The next step is to isolate the change in temperature of the blackbody radiator to χ_1. As it was mentioned earlier, the original method which demeans the training set before calculating the SVD functions as well to calibrate the HLC space. Unfortunately, this method used on the presented training set compresses the blue and green regions of a source subspace.

This is to be expected because there is less energy under the spectra of the blue, cyan and green samples than the others (see Figure 3.17). Those samples are closer to the neutral sample when the original method is used. To facilitate the study of the HLC space, the entropy minimization method was preferred as it did not have this adverse effect. The entropy minimization method was

adapted to the additional dimension as follows. The process calculates the correct rotation of one axis at a time. The following matrices are used to rotate the space about an axis

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta_x & -sin\theta_x \\ 0 & sin\theta_x & cos\theta_x \end{bmatrix}; Y = \begin{bmatrix} cos\theta_y & 0 & sin\theta_y \\ 0 & 1 & 0 \\ -sin\theta_y & 0 & cos\theta_y \end{bmatrix}; Z = \begin{bmatrix} cos\theta_z & -sin\theta_z & 0 \\ sin\theta_z & cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

X is the rotation matrix about the first dimension, Y about χ_2 and Z about χ_3. The correct rotation is determined by the minimum angle (in the event of two or more angles with equal entropy) with the minimum entropy in χ_2 and χ_3 from

$$H(\theta) = -\sum_{i=1}^{j} p(\chi_2) log_2(p(\chi_2)) - \sum_{i=1}^{j} p(\chi_3) log_2(p(\chi_3)) \quad (3.5)$$

where $H(\theta)$ is the sum of the entropy in χ_2 and χ_3 for a rotation of θ around the axis of interest, θ varies from 0 to 180° (by 0.1 increments or smaller), $p()$ is the *discrete probability distribution* (basically the image histogram). Once the correct angle is identified, the appropriate matrix from Equation 3.4 is used another time to create the affine transform for this axis and to rotate the space. The same process is repeated for the next axis on the rotated space. After the space has rotated about all three axis, the color space is calibrated to reduce the effects of blackbody radiators. The following equation is used to record the rotation about the three axes in a single matrix

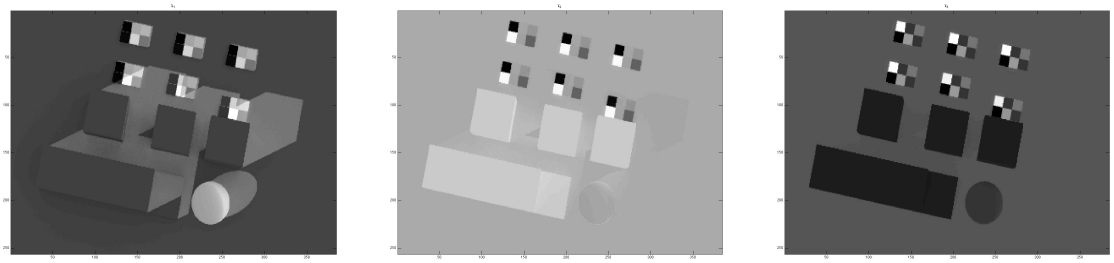
$$A = X * Y * Z, \quad (3.6)$$

this matrix allows the rotation of image data in a single step later on.

The last step is not required to use the color space, but it is useful for some application, especially the IEF or to use the HLC in a polar format. The white sample is included specifically for this purpose. The average coordinate for all white sample in χ_2 and χ_3 is calculated. This gives the location of surfaces without chromaticity under Planckian sources in this color space. Those coordinates are referred to as a White Point (WP).

3.3.4 Image Transformation

This section goes over the step to obtain HLC images from the matrices recorded in the previous section (Training and Calibration). Two matrices are required to transform an image to the HLC



(a) χ_1 (b) χ_2 (c) χ_3

Figure 3.18: The three bands of the HLC image corresponding to the 'day' version of the simulated simple scene (Figure 3.1(c)).

space using the entropy minimization method: (1) the 'input' eigenvector matrix (V) and (2) the affine transform matrix (A). The first step is to calculate the log-chromaticity differences (LCD) with Equation 3.2. If the image provided is a $x \times y \times \lambda$ data cube, then the log-chromaticity differences data cube will be the same size. In order to project the LCD onto the eigenvectors, the data cube must be reshaped to be $x * y \times \lambda$. In essence, each plane ($x \times y$), corresponds to a wavelength, must be columnized and concatenated. This is the point that requires that the training set and image cubes match, otherwise the matrix operation will not be possible. The LCD matrix is multiplied by V to obtain χ as per Equation 3.3. Only the top three bands $\chi_{1,2,3}$ are conserved, such that the data set is now $x * y \times 3$. The matrix with χ is multiplied with A and the resulting matrix is reshaped to be $x \times y \times 3$ (the HLC image). Figure (3.18) shows each band of the HLC image from the simulated simple scene.

The HLC space behaves as predicted by the theory although it is not perfect. The illumination differences are mostly confined to χ_1. The area illuminated by each source (daylight and skylight) can be easily identified in Figure 3.18(a). There are few illumination effects still visible in χ_2 and χ_3. χ_3 is the closest to being truly illuminant invariant. The image transformation code (Matlab) is in Appendix C.

3.3.5 Summary

The following diagram (Figure 3.19) contains the majority of the information required to understand the principles of the HLC space. Table 3.2 is a quick reference to differentiate the LC space

Table 3.2: LC VS HLC

	LC	**HLC**
RGB image	one dimension is illuminant invariant	reverts to LC
hyperspectral image	idem	two dimensions are illuminant invariant
day scene	functionally illuminant invariant	illuminant invariant
night scene	can have one dimension illuminant invariant	can be made illuminant invariant for at least two sources present
man-made sources	can achieve one dimension illuminant invariant	can achieve illuminant invariance for at least two sources

(Fin(2001)) from the HLC space presented in this research.

3.4 Illumination Equalization Filter

The bilateral filter does not satisfy this research's purpose. The basic assumption in this research is that if neighbouring pixels are the same color then they can be illuminated the same. The bilateral filter limits its action to the neighbourhood of the pixel of interest and checks which are similar in color. It even allows the use of different criteria of comparison in each dimension (particularly useful when using the Log-Chromaticity space). But we do not seek to average the illumination throughout the scene, we want to give each pixel the best illumination found around it. This gives three possible cases; pixels at the bottom of the dynamic range, pixels that were well exposed and highlights. Functions to deal with all three cases could have been devised, but for simplicity and time consideration, this research focused on the mid-range since it handles the majority of the typical image content (usually the content of interest). The similarity filter for the intensity is replaced by a function that gives a weight of one to pixels evenly lit, drops to zero for pixels that are darker and gives more weight to pixels that have a greater intensity calculated by

$$I(x) = \frac{\sum_{i=1}^{w} f_w(x)}{w} \tag{3.7}$$

where x are the spatial coordinates of a pixel, $f_w()$ is the value of a pixel at channel w. The new function is referred to as the discriminant function expressed by

$$u(\xi, x) = SIGN(d(I(\xi), I(x)))e^{-\frac{1}{2}\left(\frac{d(I(\xi),I(x))}{\sigma_u}\right)^{-2}} + 1 \tag{3.8}$$

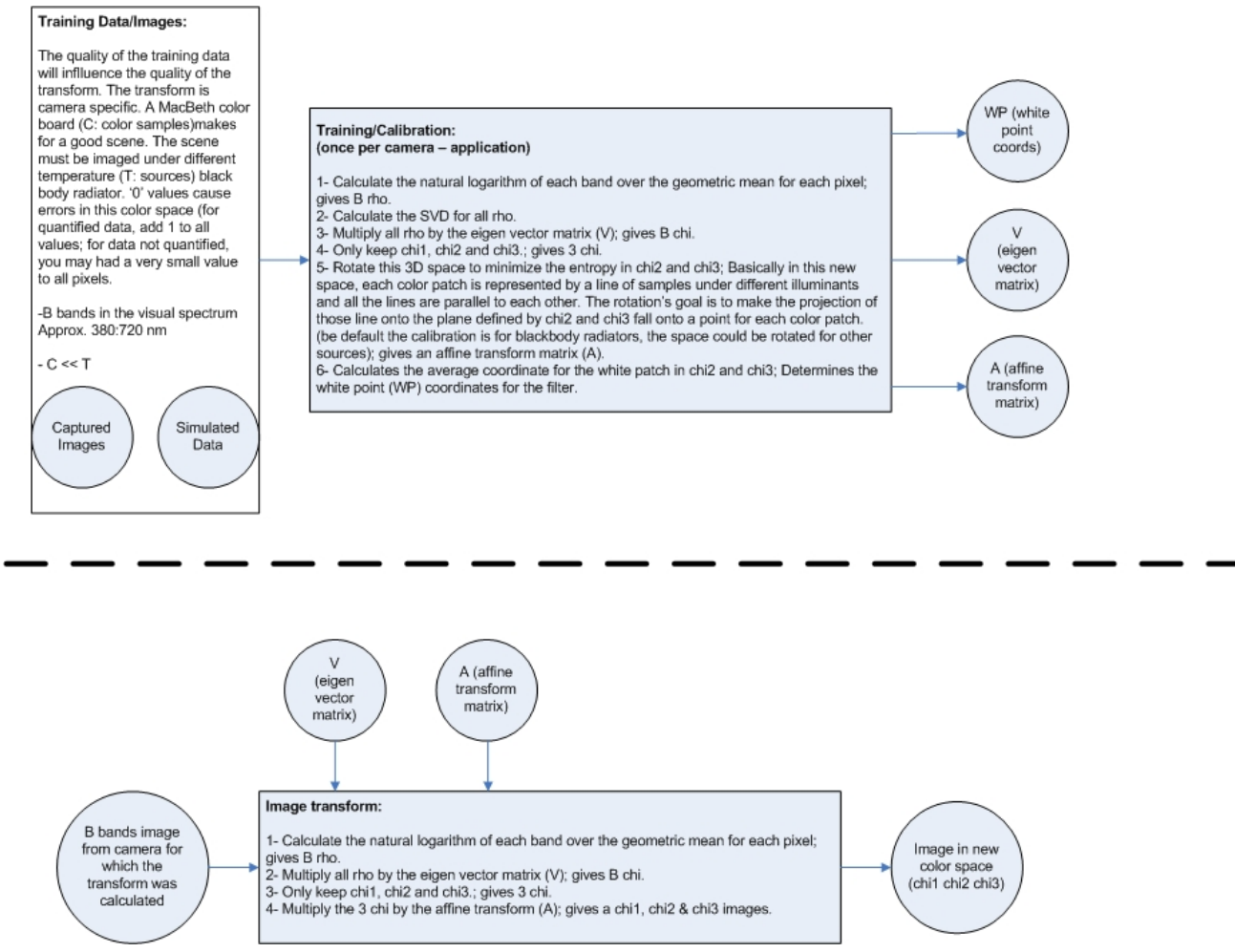


Figure 3.19: Hyper-Log-Chromaticity Diagram.

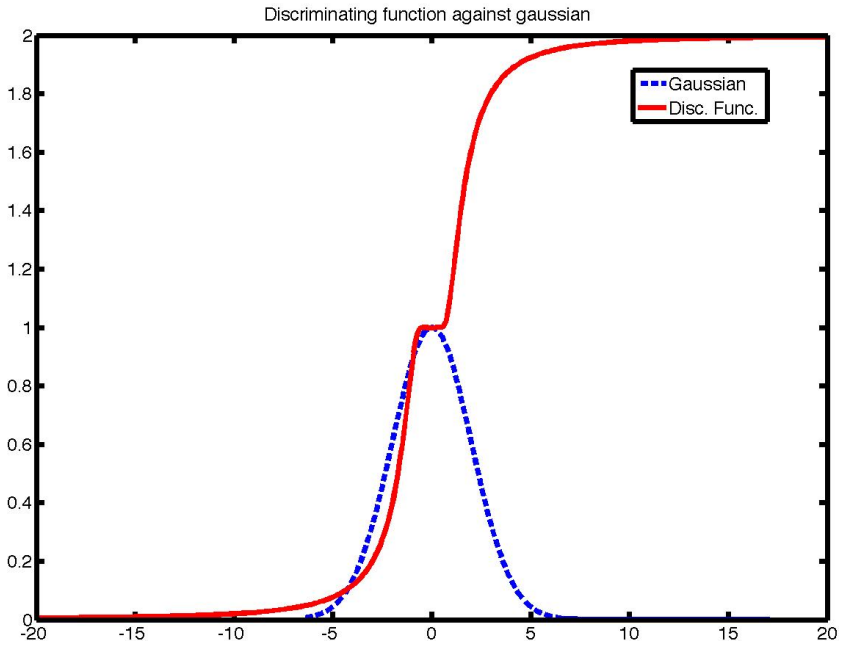


Figure 3.20: The discriminant function $u()$ over the similarity function $s()$ both with $\sigma = 2$.

where σ_u is the standard deviation of the inverse Gaussian. The discriminant function (Figure 3.20) is a biased inverse Gaussian that allows the filter to behave like a regular bilateral-filter and blur the small variation on a surface of the same color and illumination.

If those functions (similarity and discriminant) are presented with an even color and intensity patch, the Euclidian distance from the value of a near pixel with the value of the pixel of interest will be close to zero (accounting for small variations). Both filters will give a weight of one (before normalization by $k(x)$) to the neighbouring pixel exactly as the original bilateral filter would do. If the color is different, then the weight assigned by the similarity function decreases rapidly. If the intensity of the nearby pixel is greater than the pixel of interest, then the weight assigned will quickly increase to two. Inversely, if the intensity of the nearby pixel is lower then the pixel of interest, the weight assigned falls rapidly to zero. The presented filter still protects high-frequency information (changes in surface reflectance), because it retains the original similarity filter. But when the illumination varies (slowly or quickly) across a surface, it will blur with the brightest

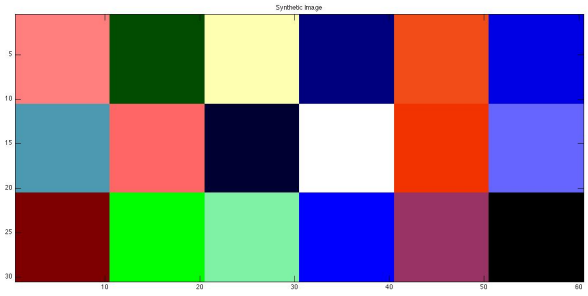


Figure 3.21: Synthetic scene designed to study the IEF.

nearby pixels. The expression for the Illumination Equalization Filter (IEF) is as follows

$$g(x) = \frac{1}{k(x)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) u(I(\xi), I(x)) d\xi, \tag{3.9}$$

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) u(I(\xi), I(x)) d\xi.$$

3.4.1 IEF Proof of Concept

To understand and verify the behaviour of the IEF, a simple function was designed with a synthetic elementary scene (see Figure 3.21). The function filters the whole image in HSI space (transform presented in Section 2.3) and allows to plot the filters for a specified pixel of interest. The scene consists of 10×10 pixel panels in a 3×6 grid. The grid is designed to have some panels of similar color near each other.

The pixel (9,21) is used to demonstrate the behaviour of the IEF because it is at the edge of a darker panel with two similar panels in its neighbourhood. The parameters for the demonstration are as follows: kernel half size of 15 pixels, $\sigma_c = 45$, $\sigma_{sH} = 1$ (standard deviation of the similarity filter in hue), $\sigma_{sS} = 0.1$ (standard deviation of $s()$ in saturation) and $\sigma_u = 0.1$ for a single filter pass. This gives a kernel of 31×31 pixels as shown in the following figures. The kernel was selected to be big enough to go beyond a neighbouring panel.

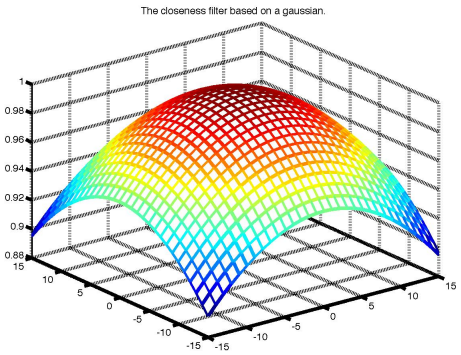


Figure 3.22: Closeness filter.

As described earlier, the closeness filter is simply a 2D spatial low-pass Gaussian and is represented by Figure 3.22. This filter does not vary with the pixel of interest and is the same throughout the image. For all the filters shown, the pixel of interest is located at (0,0), the center of the figure. Hence the centered bell shape of the closeness filter. The next filters are more interesting. The two next ones (Figure 3.23 and 3.24) correspond to the ones in Tomasi and Manduchi's original work [5]. They are the similarity filters in hue and saturation.

Figure 3.23 shows the similarity filter in hue is behaving as expected. It attributed a weight of one to the pixels of the same panel as well as to the two nearby pink panels (also a variant of red). The blue, dark blue, green and aqua panels got a weight of zero as the filter considers them different.

Figure 3.24 is the representation of the similarity filter in saturation. Saturation is more difficult to verify visually, but a verification can be made using the raw data.

Figure 3.25 is the figure of main interest in testing the IEF, since it is the new filter introduced to the bilateral filter for the research purposes. The filter behaved as expected. The cyan shelf is the panel to which the pixel belonged and all pixels were given a weight of one. The immediate neighbours are brighter and were assigned weights higher than one (in red). The pixels from the darker panels (in blue) have weights below one.

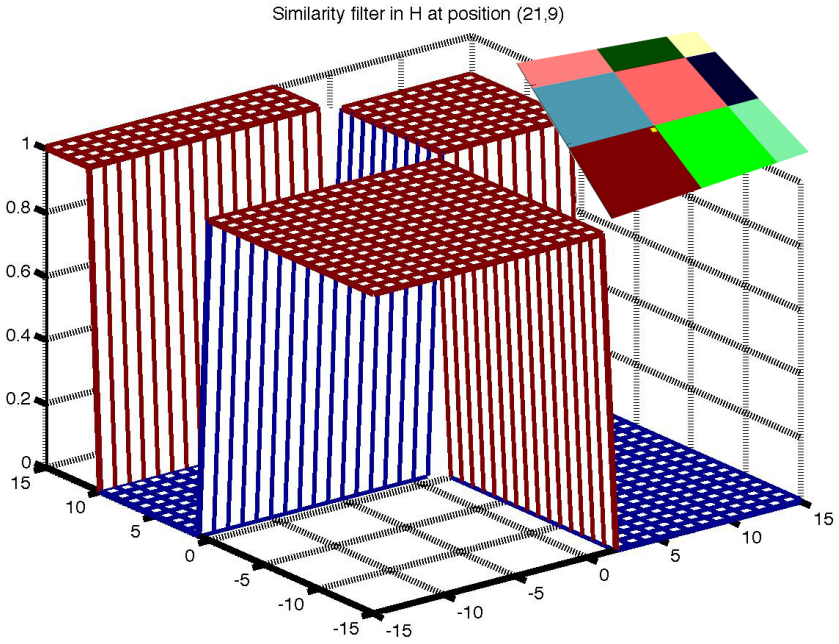


Figure 3.23: Similarity filter in hue (pixel (21,9)).

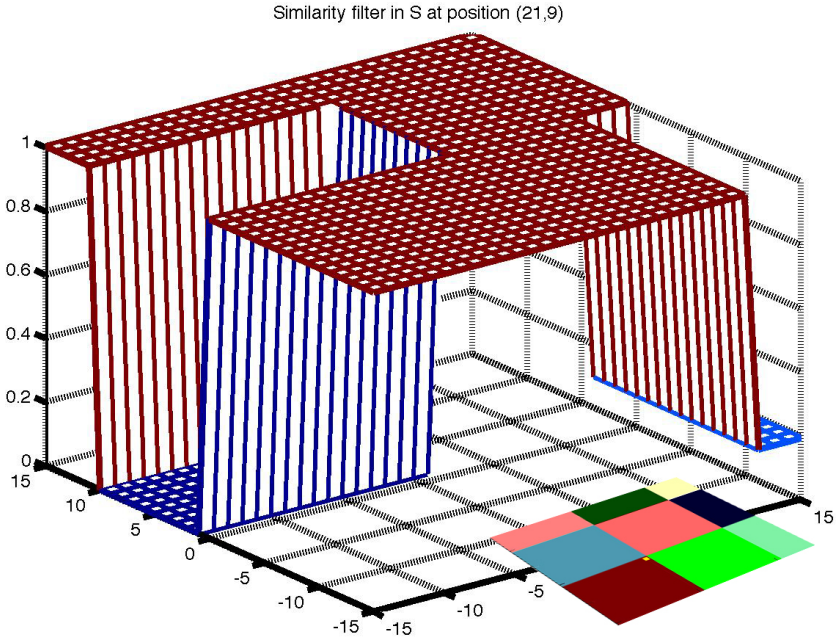


Figure 3.24: Similarity filter in saturation (pixel (21,9)).

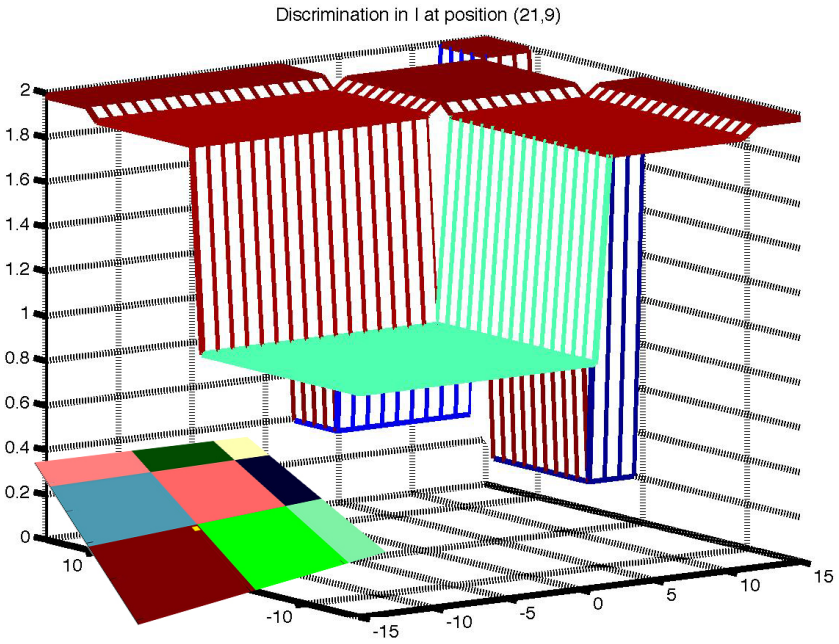


Figure 3.25: Discriminant in intensity (pixel (21,9)).

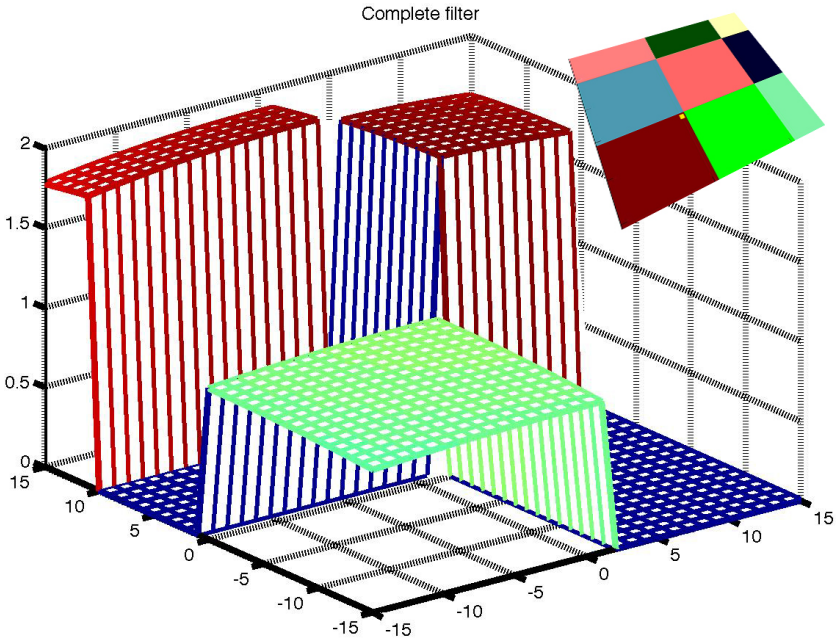
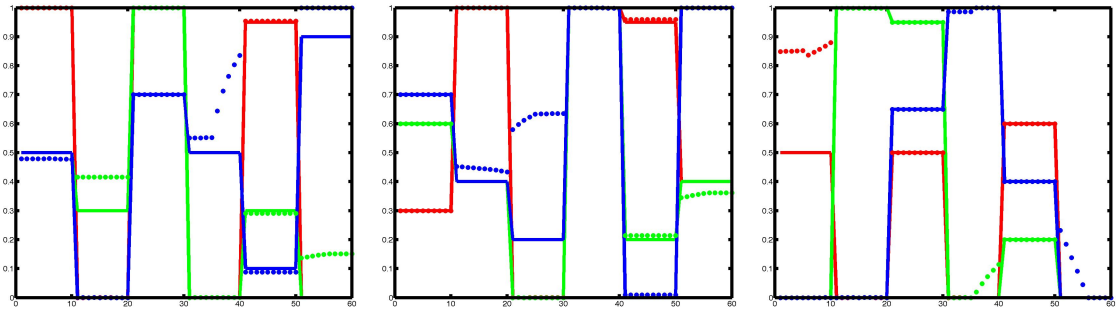


Figure 3.26: Complete filter.



(a) Row 7 (b) Row 15 (c) Row 25

Figure 3.27: Three slices from the original (plain) and filtered (doted) images.

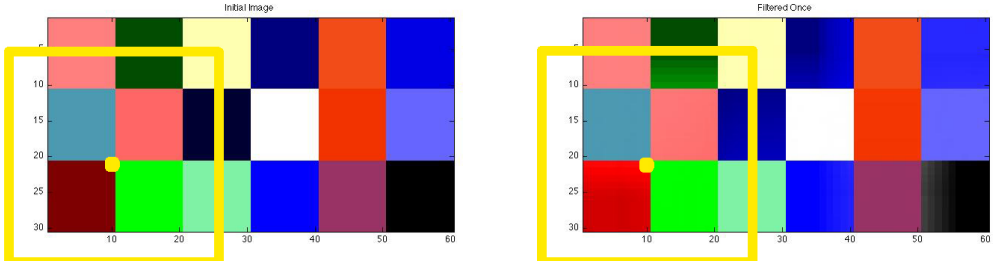


Figure 3.28: Left: Original synthetic image. Right: Filtered once result. Each image has the pixel of interest and the kernel of the previously shown filters marked.

Figure 3.26 is the resulting filter when all the previous filters are combined before it is normalized. Only the pixels with red hue have a weight above zero (as the filter for the similarity in hue). The curve of the spatial Gaussian is visible, giving slowly (because σ_c is rather big) decreasing weights as pixels are further from the pixel of interest. The area of even intensity with the pixel of interest has a weight close to one while the brighter pixels are given a weight close to two. The image product should result in a brighter panel in the bottom left of the grid.

Figure 3.4.1 contains three graphs, each a horizontal slice through both image. The plain lines are from the original and the doted lines from the filtered image. The line color matches the image channel. This shows quantitatively the impact of the filter on the image.

The last figure (3.28) of this section shows the IEF works as designed. It brightened the darker

pixel that had a nearby brighter panel of a similar color without blurring spatial features. Panels of different color were not affected. The black panel is brightened by the white panel which is due to the color space used. HSI gives the same hue and saturation to all neutral colors and differentiates them only by intensity. Hence white is a brighter black and the error is not a fault of the filter. Although the orange panels don't seem to change, they are actually changing slowly. The discriminant filter is based on the Euclidian distance of the intensity of pixels, small differences will incur small changes. This image is the result of a single pass of the IEF. With multiple passes the illumination gradients would be replaced by even illumination and all originally similar panels would be exactly the same.

3.4.2 Parameters

The IEF requires the original image (img) as input and user specified parameters. The first one is the kernel's half-size rounded down (e.q., 5 gives a kernel of 11 by 11)(how many pixels away from the pixel of interest), followed in this order by: the standard deviation for the spatial Gaussian filter (σ_c), the standard deviation of the similarity filter for the first chromaticity dimension (σ_{sH}) (hue in the proof of concept), the standard of the similarity filter for the second chromaticity dimension (σ_{sS}) (saturation in the proof of concept), the standard deviation for the discriminant filter (σ_u), the highlights cutoff (HL) and the shadows cutoff (SL). HL and SL are parameters to prevent the filter from acting on pixels for which it was not designed to enhance. HL and SL can take any value between zero and one. HL allows pixels above the upper limit ($HL * max[I(x)]$) to bypass the filter while SL allows pixels under the lower limit ($SL * max[I(x)]$) to do the same. By setting HL to one and SL to zero, those parameters have no effect.

3.4.3 Filtering

The present implementation of the IEF starts by creating images with mirrored edges based on the image to be filtered and the corresponding HLC image. The vertical edges are mirrored first. The width of the mirrored edge is determined by the kernel's half-size rounded down. Then the horizontal edge (including the pixels added to each side of the image) are mirrored. This way, the edge of the image will not impact the filtering process (aliasing). The image is processed sequentially starting with the top left pixel of the original image within the image with edges

mirrored. This version of the IEF steps through columns a row at a time.

3.5 Merging of HLC and IEF

The algorithm combining the HLC and the IEF requires a few extra inputs and parameters compared to the IEF alone. It requires the original image (img), the corresponding HLC image (LCimg) and the WP (White Point) of the LC space as input. It also has a number of user parameters. The first one is the kernel's half-size rounded down (as the IEF), the standard deviation for the spatial Gaussian filter (σ_c), the standard deviation at the WP (σ_{WP}), the standard deviation away from the WP in χ_2 (σ_{χ_2}), the standard deviation away from the WP in χ_3 (σ_{χ_3}), the standard deviation for the discriminant filter (σ_u), the highlights cutoff (HL) and the shadows cutoff (SL). Three standard deviations are required for the similarity filter because of the nature of the LC space that is much like any hue and saturation space (ambiguity near the 'centre'). Using polar coordinates would not have been sufficient to accomplish the same result. Instead the standard deviation of the similarity filter ($\sigma_s(x)$) is varied in function of the Euclidian distance of the value of the pixel of interest from the WP:

$$\sigma_s(x) = \frac{\sigma_\chi - \sigma_{WP}}{max[|d(LCimg, WP)|]} \left| d(LCimg(x), WP) \right|. \tag{3.10}$$

The following diagram (Figure 3.29) is a summary of everything discussed in this section and the Matlab code is found in Appendix D.

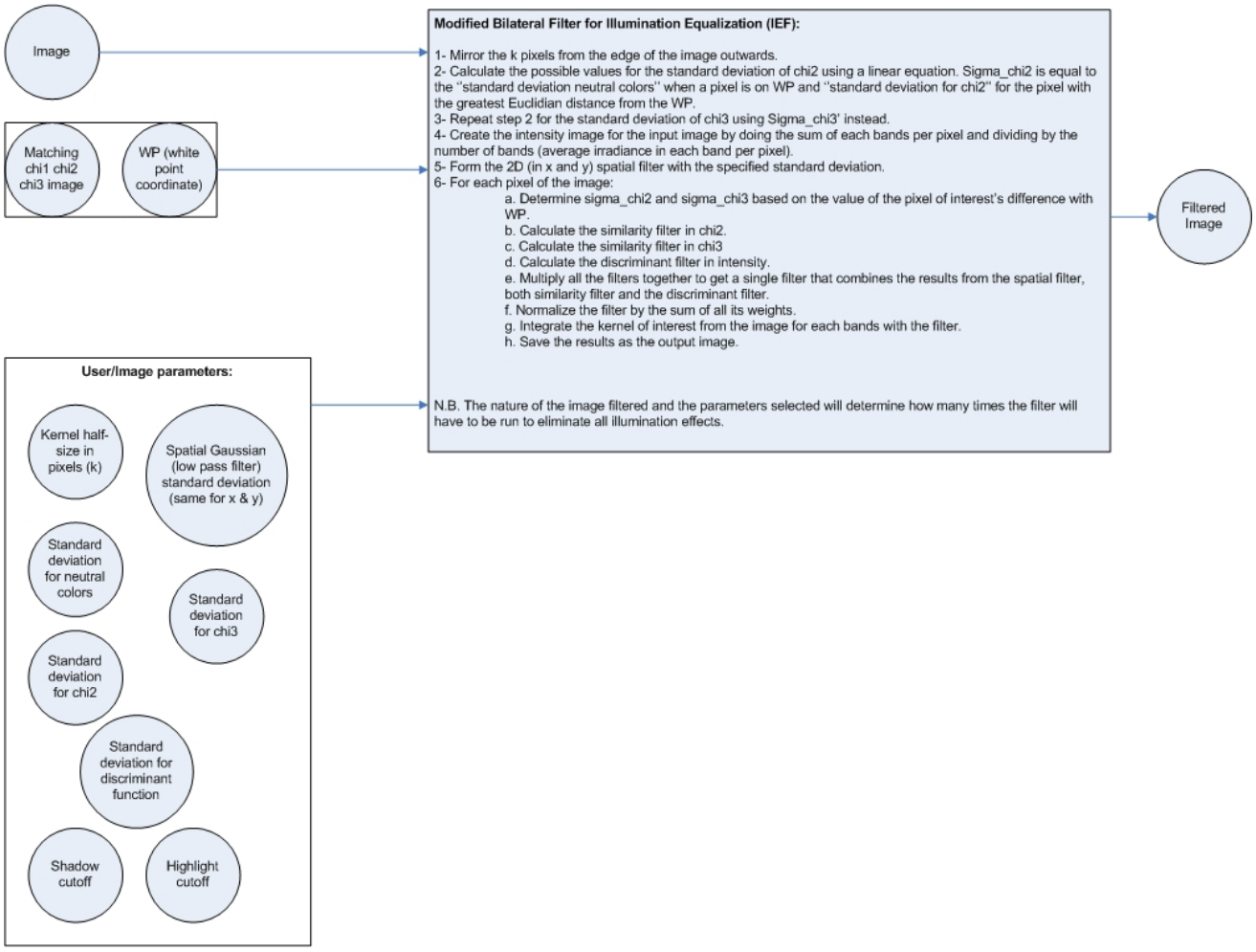


Figure 3.29: Filter diagram using the HLC space.

Chapter 4

Results and Discussion

This chapter begins with all the images from the various experiments that were generated to understand and evaluate the work of this research. The images are followed by the compilation of the statistics of interest. The last section is where the presented research is discussed.

4.1 Experiments

The experiments are a progression from small, controlled data sets to more complex, realistic image simulations. The first experiments could be studied closely in very little time to ensure the algorithm functioned properly before larger data sets were used which would be more time consuming to study. The first experiments were also designed to be easy to study. The desired outcome was known for every pixel and made to be easily differentiated from the initial state. Later experiments were not focused on each pixel value, but rather on the behaviour of the algorithm with different images. The progression starts with synthetic images, followed by simple simulated images and ends with complex simulated image. Although all images are essentially synthetic, only the first set is not entirely based on recorded spectra. The experiments with those images were meant to study the algorithm itself, while the last experiment was meant to quantify the results of this research.

Classifiers (Section 2.7.1) were readily available and are known applications with a number of metrics of success (Section 2.7.2). In the context of surveillance, the desired result of a classifier is

to properly identify the materials in the scene (an intermediary step of a surveillance application). The goal of the last experiment was to measure how much better (at identifying materials) the proposed algorithms did against the untreated image and a common improvement method (Section 2.7.3).

While this research focused on proving the feasibility of the proposed idea, it did so with a realistic model albeit perfect. These experiments do not consider a number of limitations. DIRSIG sampled a 3 by 3 grid for each pixel and photons were permitted to reflect once. Their was no actual sensor considered, so the impact of sensor spatial or spectral resolution, pixel size, noise, dynamic-range, etc. were not considered throughout the experiments. The impact of path radiance and the nature of the medium (haze, aerosols, water content, etc) was not included. Basically the experiments considered perfect recording and each pixel is the equivalent of a measure *in situ*.

4.1.1 Synthetic

There are no recorded results from the experiments with the synthetic images. The early results were viewed on screen to adjust the proposed algorithm. Once the algorithm functioned as desired, these images were discarded. Later experiments with simulated scenes led to further modifications of the algorithm. Those modifications were significant enough that the experiments with synthetic images would need to be updated to use the new functions. Since the latest experiments were proceeding as expected, there was no motivation to go back and fix older functions.

As it was mentioned in Section 3.1.1, the synthetic experiments were the simplest in nature to facilitate the visual assessment of results. The flat grid of panels was lit with different sources, resulting in a finer vertical grid. The objective was to get an output closer to the original grid. Not the same color, but rather the equivalent under the brightest illuminant present. Figure 3.1.3 was the first step, it showed the algorithm worked with the illuminants it was trained with and calibrated for Section 3.3.3. The second step used the radiances propagated through the atmosphere (Figure 3.4), the realistic natural illuminants. The algorithm was expected to work; the illuminants did not have the same coordinates than the Planckian sources, but they were closely clustered with each other (Figure 3.14). This required the algorithm to be a bit more flexible with the similarity

criterions, but it still was able to recover the grid without adverse effects. The last step was to deal with illuminations for which the algorithm was not calibrated and which were spread out in HLC space (Figure 3.5). The results were poor as expected, but two simple white balance scheme were implemented with this image. The effect of different calibrations and the translation of coordinates in HLC space was tested. Those two schemes proved successful even though they are impractical for most applications (requires additional or prior knowledge of the scene).

The good results with the highly predictable and simple to study experiments gave confidence in the potential of the algorithm and permitted to go onward.

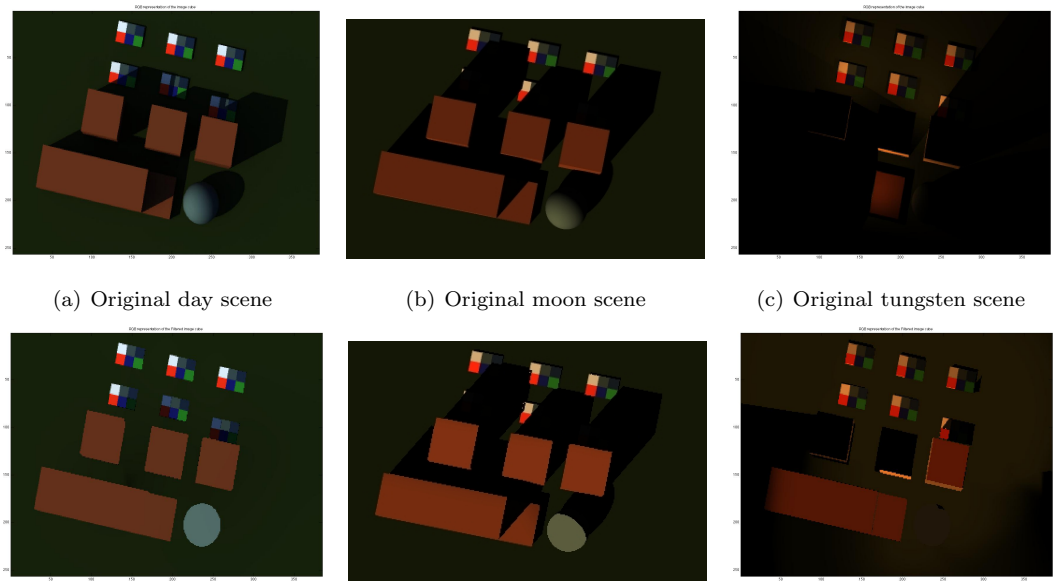
4.1.2 Simple Simulation

The simple simulation introduces measured spectra for real materials, tridimentional geometry, a curved surface and some radiance path considerations (direct and downwelled radiance, adjacency effects) for natural and artificial sources. Thus light falloff, shadows and soft edges can be observed in these experiments. The synthetic images only had hard edges; a given pixel was of a set material illuminated by a set source, the result was known with absolute certainty. Soft edges are due to spatial mixes in material and illumination. Even though these experiments add all those elements, there are few enough features present in the scene such that the study of results is straightforward. The images labeled 'Filtered' are the results from the algorithm combining the HLC and the IEF.

Visually, Figure 4.1(d) is a very good example of the expected effect of the proposed algorithm. All light falloff effects have been removed and every pixel of a surface has the same illumination. All surfaces do not have the same illumination because the well lit surfaces were not close enough to the surfaces in shadows. In the case of this simple scene, the kernel size of the filter could have been made big enough to equalize the illumination of all panels. However it is unlikely that a real scene would allow for very large kernels.

The filter had very little effect on the 'moon' scene (Figure 4.1(e)). The projected light effects were removed, but its sources are very different in HLC space. The moonlit sky is nearly Planckian while moon light is more red. Its shadows were not affected by the filter.

The 'tungsten' scene (Figure 4.1(f)) is also a good example of the potential of the proposed solution. The grass and large brick block are in the process of being entirely retrieved from the shadows (the filter was ran 40 times or less on all images, this scene could have used a few more



(a) Original day scene (b) Original moon scene (c) Original tungsten scene

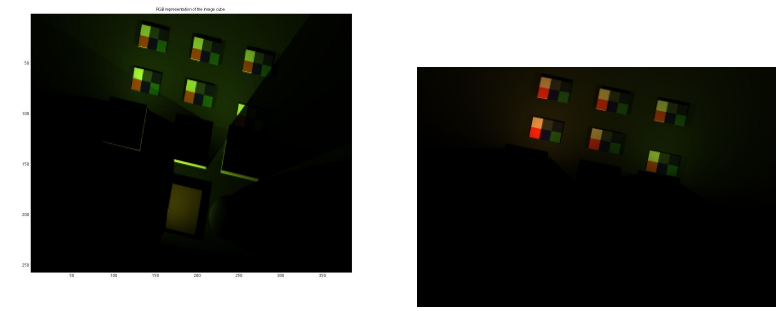
(d) Filtered day scene (e) Filtered moon scene (f) Filtered tungsten scene

Figure 4.1: Simple Simulated Scenes (part 1).

passes). Depending of the monitor/display method, it would seem that the sphere is being changed to grass. The side of the small blocks are not being recognized as the same material of the rest of the block because of the highlight cutoff. Some shadows (around the panels) are not being affected by the filter.

Variations of intensity are being removed in the 'sodium' scene (Figure 4.2(b)). The sources causes color confusion between some materials. The asphalt sphere is turned to grass and the top right blue panel is turned to asphalt. As for the 'tungsten' image, the side of the blocks are not recognized as the same material as the top of the blocks. The grass lit by the sodium source is progressively made brighter.

The 'mts' (moon-tungsten-sodium) (Figure 4.2(d)) result is particularly interesting, it clearly demonstrates the behaviour of the filter with different sources present. The other scene did not offer the same opportunity because the artificial sources were all more powerful than the natural sources. In this case, the artificial sources are close to the same power. The most powerful illuminant (brightest) on a pixel is the one favored by the algorithm, this is observable by the straight line separating the lit area in two between (the tungsten is a bit stronger then the sodium



(a) Original sodium Scene (b) Original mts Scene

(c) Filtered sodium Scene (d) Filtered mts Scene

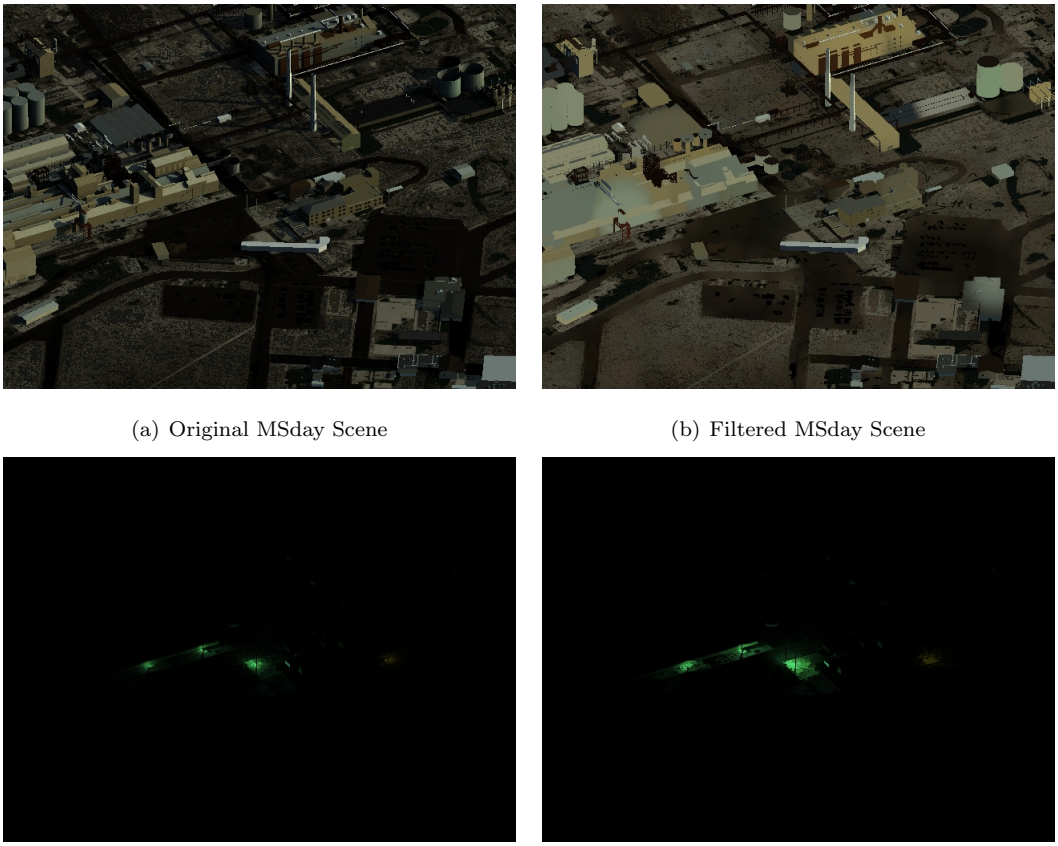
Figure 4.2: Simple Simulated Scenes (part 2).

sources) the two sources. Another interesting observation is that there is no confusion on the panels under the sodium illuminant in this case.

4.1.3 Megascene2

The Megascene2 portion is meant to be a more complex test and the results are also more complicated to interpret. These experiments are representative of a realistic industrial surveillance plan scenario. The filter had some desired effects (Figure **??**): it removed most shadows and all projected light effects, objects are now identifiable in the parking lots (simulation artifacts where cars should be found, not an algorithm error), most structure are unicolor. It also had some undesired effects: the windows of some building disappeared, there are growing regions that seem to be caused by a minority pixel.

Overall for night scenes, the proposed algorithm equalized the areas under an illuminant and made more of the scene brighter. This is also observable in the night version of the Megascene2 portion (Figure 4.3(d)).



(a) Original MSday Scene

(b) Filtered MSday Scene

(c) Original MSnight Scene

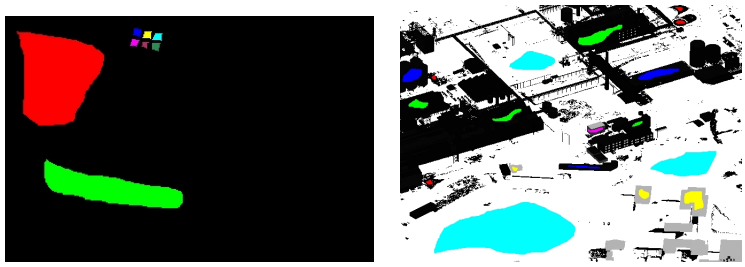
(d) Filtered MSnight Scene

Figure 4.3: Scenes from a portion of Megascene2.

4.1.4 ROI and Truth Maps

This section shows the Region of Interest (ROI) that were used to train the supervised classifier and the truth maps that were used to calculate the confusion matrices. The ROIs were selected from the material images (see Figure 4.1.4). If an image is favored by the ROIs, it is entirely unintentional.

ROI

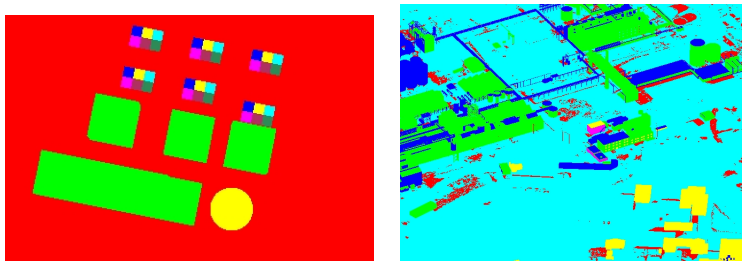


(a) for the Simple Simulated Scene (b) for the Megascene2 portion

Figure 4.4: ROI to train the supervised classifier.

Truth Maps

This is the legend for the truth map of the simple simulated scene (Figure 4.5(a)): red = grass, green = red brick, yellow = asphalt, blue = green roof shingle, cyan = panel painted blue, magenta



(a) for the Simple Simulated scene (b) for the Megascene2 portion

Figure 4.5: The truth maps for each scene.

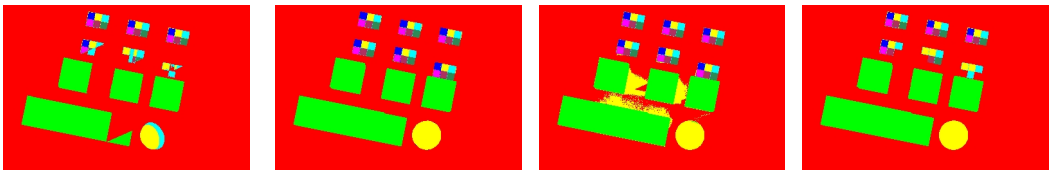
= panel painted red, maroon = panel painted with blue car paint and kaki = panel painted green. The six material classes from the portion of Megascene2 are not known.

4.1.5 Classmaps

This section contains all the classmaps created to assess the potential of the HLC space and the algorithm combining the HLC and the IEF. 'Original' is the original image as rendered by DIRSIG (see Figure 3.1). 'SUN' stands for Spherical Unit Normalization, the original image is spectrally normalized (explained in Section 2.7.3) and serves as a benchmark method for the reduction of illumination effects. 'HLC' is for the two band image obtained from the HLC transform (see Section 3.3.4). 'Filtered' is for the result of the proposed algorithm (Section 3.5). The section starts with the results from the supervised classifier followed by the unsupervised classifier.

Minimum-Distance

The classmaps resulting from running Minimum-Distance (explained in Section 2.7.1) in ENVI [19] using a single set of ROIs (Figure 4.1.4) for all images. The first set of classmaps is from the simple day scene (see Figure 4.6).

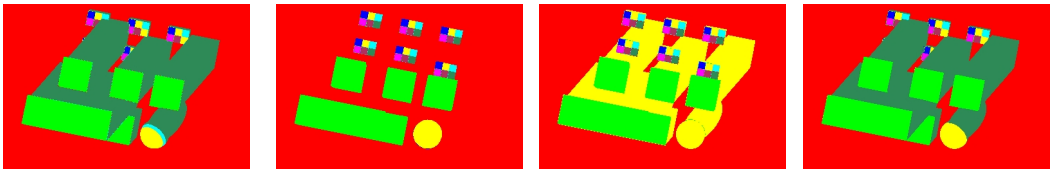


(a) Original (b) SUN (c) HLC (d) Filtered

Figure 4.6: Minimum-Distance: day.

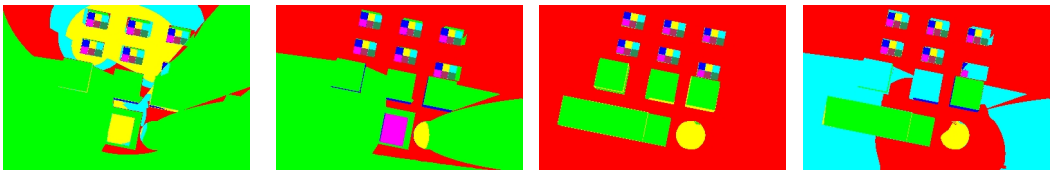
The classmap from the original day image (Figure 4.6(a)) is a good example of typical issues with classifiers. The light falloff causes the sphere to be split in three classes. All shadow pixels are misclassified, most of them are confused with the grass class. The SUN image performed very well. The HLC classmap has issues with light reflected off of two brick surfaces (yellow patches between blocs). The filtered classmap has misclassified the panels that are still in shadows, but it has removed shadow edges. Pixels from a panel are all assigned the same class.

The classmaps from Figure 4.7 are from the simple night scene with moon light.



(a) Original (b) SUN (c) HLC (d) Filtered

Figure 4.7: Minimum-Distance: moon.



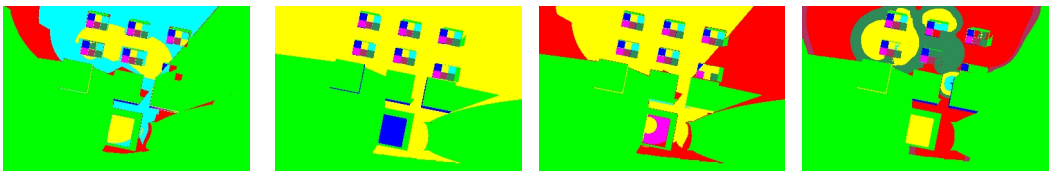
(a) Original (b) SUN (c) HLC (d) Filtered

Figure 4.8: Minimum-Distance: tungsten.

The original moon classmap (Figure 4.7(a)) looses all information in the shadows and also splits the sphere in two classes. The SUN classmap does well for this scene also. The HLC classmap was not able to remove shadows (the moon not being a Planckian source), but was able to recuperate all panels and consider the sphere as a single object. The only benefit of the filtered classmap over the original, it is that the light falloff has been removed.

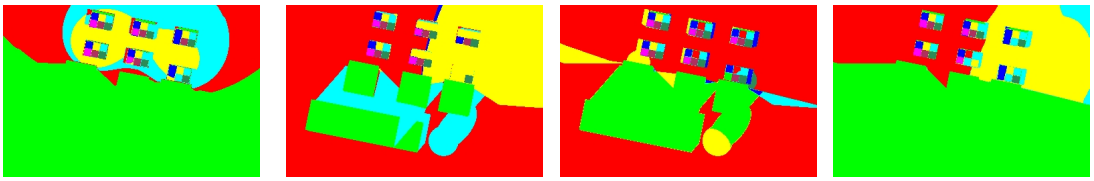
The classmaps from Figure 4.8 are from the simple night scene with a tungsten source.

The original tungsten classmap (Figure 4.8(a)) has five recognizable groups of panels and the rest is all misclassified (bricks with grass with asphalt). The variations in illumination are easily identifiable on this classmap as well as the location of the artificial source. The SUN classmap removed most variations in illumination but the location of the sources is still identifiable. The panels are accurately classified, but there is confusion between brick, grass and shadows. The HLC accurately classifies the majority of the scene. There are a few pixel in error on the top of the sphere and the side of the blocs facing the light source. The filtered classmap could have used a few more additional passes of the filter to finish growing grass region at the bottom. Two of the blocks and the sixth group of panels are not being recuperated by the algorithm due to the spatial limitation of the filter. Visually, the sphere was almost impossible to identify, but the classifier is



(a) Original (b) SUN (c) HLC (d) Filtered

Figure 4.9: Minimum-Distance: sodium.



(a) Original (b) SUN (c) HLC (d) Filtered

Figure 4.10: Minimum-Distance: mts.

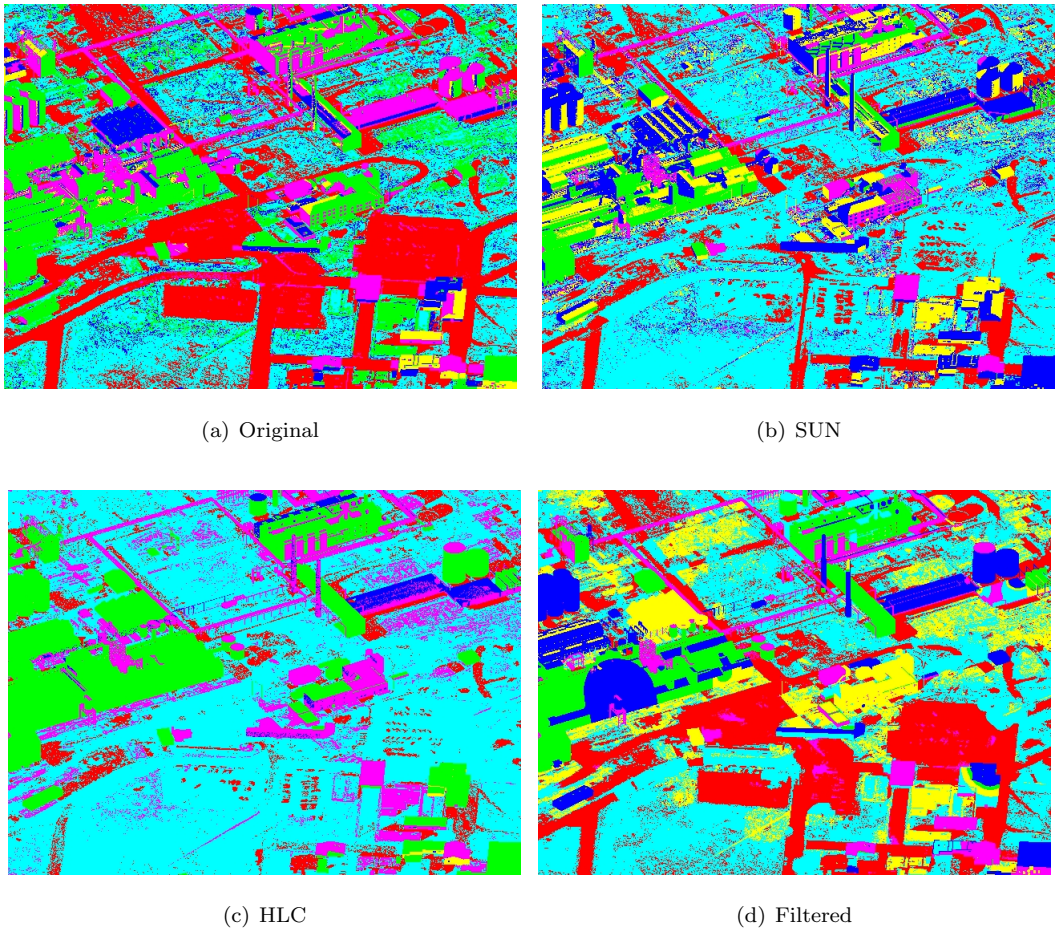
still able to pick it out.

The classmaps from Figure 4.9 are from the simple night scene with a high pressure sodium source.

The original sodium classmap (Figure 4.9(a)) is very similar to the Original tungsten classmap, the artificial sources are in the same position. There is one small difference in the classification of the bottom right group of panels. Two more panels are identified in the shadow but they are misclassified. The SUN classmap has the same shape as the tungsten SUN classmap but no pixels have been assigned the grass class and the block under the source changed from red panel to green roof shingle. The HLC and Filtered classmap are mostly misclassified. The HLC classmap managed to recuperate five groups of panels. The filtered classmap still has a growing grass region at the top but the majority of the regions have nothing to do with the truth.

The classmaps from Figure 4.8 are from the simple night scene with a tungsten source and a high pressure sodium source.

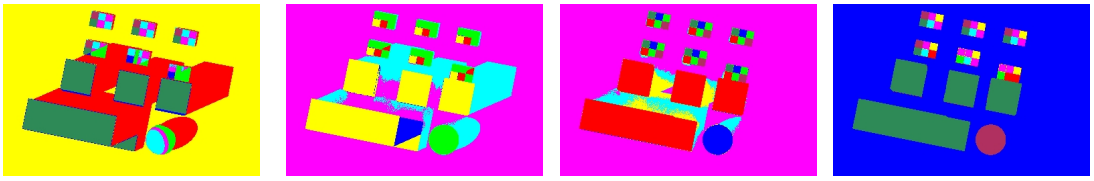
Of all four classmaps from the mts scene (Figure 4.10), HLC is the least affected by the two artificial sources present, but SUN is the only one to separate the blocks from the grass.



(a) Original (b) SUN

(c) HLC (d) Filtered

Figure 4.11: Minimum-Distance: MSday.



(a) Original (b) SUN (c) HLC (d) Filtered

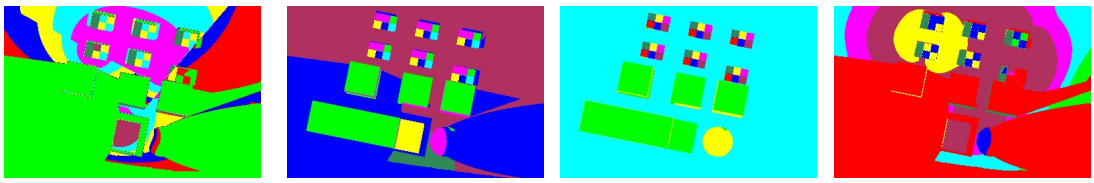
Figure 4.12: *k-means* - day.

The classmaps from the Megascene2 portion (Figure 4.11) might be too complex to properly interpret visually, there are a few observations to be made. The original classmap is the only one affected by projected light effects. Both the HLC and filtered classmaps are shadowless as can be easily picked up by looking at any of the cylindrical structures. The filtered classmap has one obvious region of confusion (center-left of the image) where one building class is slowly overtaking the other. It is also attributing most road pixel to the wrong class.

k-means

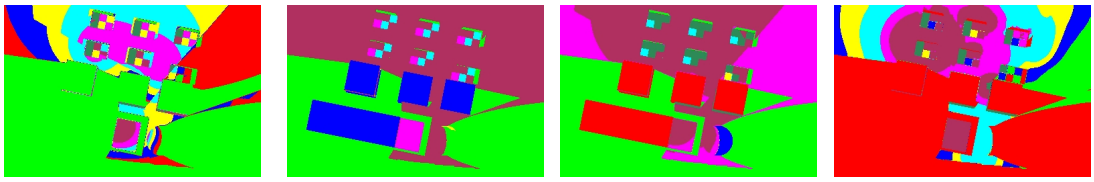
This section covers the classmaps resulting from running *k-means* in ENVI [19] (8 classes, 0.01% change threshold, limited to 20 runs). Since *k-means* is unsupervised, the interpretation of the classmap is a bit more complicated. The colors of the classes do not necessarily correspond, so attention must be paid to the nature of the class rather than its color code. Since the classifier was asked to find eight classes, any subdivision of a truth class will also mean another truth class will be mis-classified.

The HLC classmap of the day scene (Figure 4.12) is the only one not confused by pixels in shadows, but it is also the only one mis-classifying pixels with adjacency effects. It attributed a class to grass pixel with a single reflection from brick and another to grass with two reflections from brick surfaces. It merged the red painted panel class to the brick class and the two variations of blue painted panels. The original classmap is confused by shadows and light falloff effects as anticipated. The SUN classmap is confused by pixels in shadows and adjacency effects. The filtered classmap is still confused by pixels in shadows but a great number of these have been replaced by their well lit equivalent, so it is overall the best (visual) result for this specific experiment.



(a) Original (b) SUN (c) HLC (d) Filtered

Figure 4.13: *k-means* - tungsten.

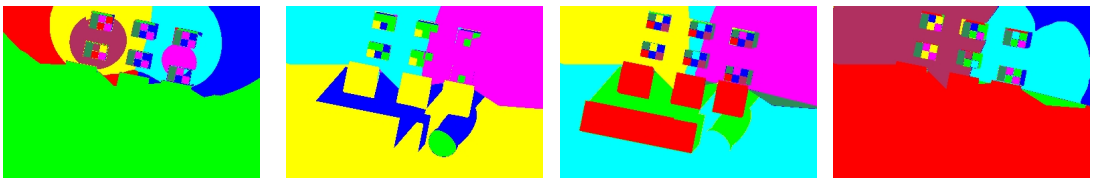


(a) Original (b) SUN (c) HLC (d) Filtered

Figure 4.14: *k-means* - sodium.

The majority of the classes (see Figure 4.13) were attributed to variations in illumination over grass pixels in the original image, the only exception being the kaki class (mixing roof shingles and red painted panels). The SUN classmap has similar results, it split both the brick and grass class into three variation of illumination respectively, which only left two classes for the panels. The HLC classmap is almost error free, there is a small group of brick pixel on the (asphalt) sphere and the brick sides directly illuminated by the sources are confused with asphalt pixels. The filtered classmap did marginally better than the original image, regions of even illumination are growing, the smaller areas are mostly under one illumination, but the large surface of grass could have used many additional passes of the proposed algorithm.

The classmap of the original image with the sodium source (see Figure 4.14) is very similar to the result of the tungsten scene; the classes represent change in illumination instead of changes in material as desired. The SUN and HLC image did not find eight separate classes and divided the brick and grass classes. The filtered result is work in progress (it was limited to 30 runs) much like the tungsten classmap with growing regions of illumination (which should ultimately result in one area of constant illumination for the given material).



(a) Original (b) SUN (c) HLC (d) Filtered

Figure 4.15: *k-means* - mts.

The presence of two sources can be observed in all classmaps of the mts scene (Figure 4.15). The original image is the only one that is confused by projected light effects. The SUN images only attributed five classes. The HLC classmap is the closest to presenting a result similar to the truth map.

4.2 Statistics

The following tables are the compilation of a subset of the statistics provided by the confusion matrices in ENVI. Table 4.1 handles the results from the supervised classifier on the left and the results from *k-means* on the right. The results are grouped by scene. The classmaps obtained from the classifiers are presented in Section 4.1.5. "Original" is the unprocessed spectral irradiance image provided by DIRSIG, "SUN" is the image spectrally normalized, "HLC" is a 2 band image made of χ_2 and χ_3 (see Section 3.3.4) and "Filtered" is the result of the combination of the HLC and the IEF (see Section 3.5). The kappa coefficient is provided by the confusion matrix ($\hat{k}$). "Sum error" is the sum of all the commission and omission percentages. It is an indicator of the number of errors made by the supervised classifier. "Class%" is the number of pixels considered by the confusion matrix over the number of pixels in the image. Since *k-means* is unsupervised, the classes it identifies could be completely different from the truth classmap. As mentioned previously, classes can be divided or merged by the classifier. The confusion matrix functions by associating a class to a truth class. If the classifier splits a class into two classes, then only one of the two can be associated to the truth class and the other is not considered in the statistics. "Class%" is an indicator of how much of the image the statistics are considering. The MSnight scene was considered to have to few identifiable features to be classified. The moon and MSday scene had

poor results with the supervised classifier and were not submitted to the unsupervised classifier.

Without an actual specific purpose the statistics hold little value. The value is determined by the importance of the decision based on those statistics. Is a kappa coefficient of 0.9 sufficient to take action? The decision will depend on the action, sending someone to get a visual is not as critical as bombing a target. In the context of this research, the relative value of $\hat{k}$ is more important then its absolute value; the methods improving or degrading the classifiers accuracy compared to the untreated result.

Focusing on the results from the Minimum-Distance classifier first, the original day scene is the best $\hat{k}$ (0.89) amongst all original images, leaving little room for improvement, arguably the hardest 0.13 to improve (harder to reach 100% accuracy then 13% accuracy). The other methods still manage an improvement of at least 0.04 which is not bad, but the real improvement occurs in the reduction of errors. The other methods roughly cut the *sum error* by half. For the day scene, the best method is the SUN image. It is hard to assign the second position as the application would determine wether the accuracy or number of error is more important. The HLC image achieved a lower $\hat{k}$ and *sum error*. Since *sum error* is not as well understood/popular, the assumption is that accuracy is more important so the filtered is second and the HLC image third.

The night scenes are clearly harder to classify by the Minimum-Distance. The original moon image has the hight $\hat{k}$ value (0.55), a loss of 32% of accuracy. The SUN method gets the best results for this case also. It happens to be the case for which SUN is a drastic improvement over all other compared methods both in accuracy and number of errors. The HLC image brings a small improvement in accuracy and greatly reduces the number of errors, it is the second best method for the night scene. Filtered marginally improves on the original results and could almost be considered equal to it. Filtered and original are in third position.

The tungsten scene allows the proposed methods to shine. Even though the SUN method improves the accuracy by 0.22 and the *sum error* by 243.25, the filtered images improves the accuracy further by 0.12 (and a small improvement in errors) and the HLC image achieves an amazing 0.96 accuracy and cuts the errors to less then a fifth, making it the highest improvement from all experiments.

The sodium scene is the most difficult for all methods with the worst results. The HLC method

Table 4.1: Classifier statistics summary.

		Minimum-Distance		***k-means***	
	Looking for:	HIGH	LOW	HIGH	HIGH
day		$\hat{k}$	**sum error**	$\hat{k}$	$\hat{k}$***Class%**
	Original	0.87	324.03	0.89	0.74
	SUN	0.98	68.94	0.99	0.84
	HLC	0.91	124.87	0.98	0.91
	Filtered	0.96	185.61	0.95	0.94
moon					
	Original	0.55	492.98		
	SUN	0.98	70.64		
	HLC	0.62	193.01		
	Filtered	0.56	428.84		
tungsten					
	Original	0.09	573.73	0.12	0.10
	SUN	0.31	330.48	0.35	0.34
	HLC	0.96	115.08	0.96	0.96
	Filtered	0.43	305.11	0.14	0.12
sodium					
	Original	0.07	648.61	0.12	0.12
	SUN	0.10	547.02	0.31	0.30
	HLC	0.16	486.90	0.47	0.37
	Filtered	0.12	794.83	0.10	0.09
mts					
	Original	0.16	519.51	0.11	0.10
	SUN	0.43	586.15	0.21	0.17
	HLC	0.68	355.71	0.52	0.37
	Filtered	0.17	562.80	0.18	0.15
MSday					
	Original	0.26	380.53		
	SUN	0.43	593.98		
	HLC	0.59	410.44		
	Filtered	0.29	635.52		

bests the others in accuracy and number of errors, the SUN and filtered classmap are both more accurate but filtered has more errors then the original; so this puts the SUN method second and filtered third for this scene.

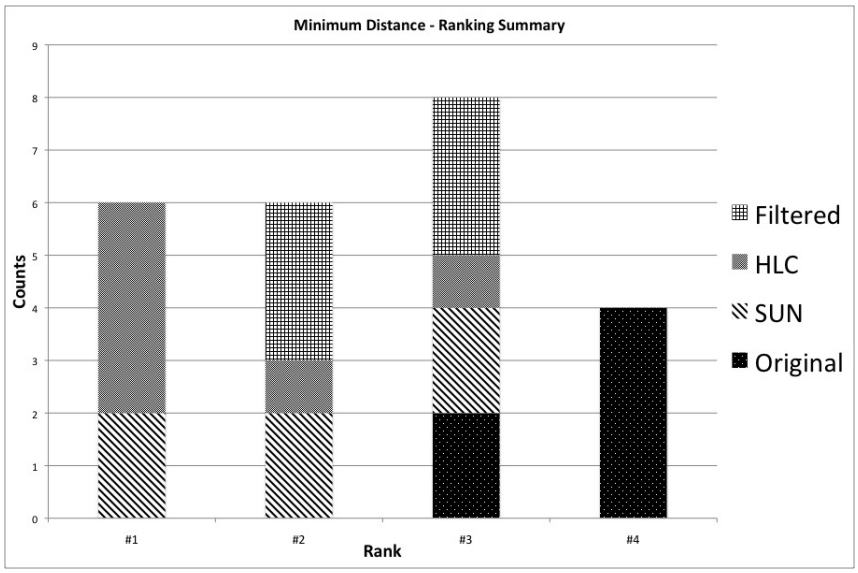
The Filtered classmap is equivalent to the original result, having a slight increase in accuracy and a moderate increase in errors. The HLC has the best results, more then tripling the accuracy and cutting one third of errors. The SUN method is second, doubling the accuracy but doing more errors.

The HLC is the best method for the MSday scene, followed by the SUN method. Even though the accuracy is the primary factor of performance for this research the number of errors done by the filtered method is perhaps enough to consider it brought no real improvement.

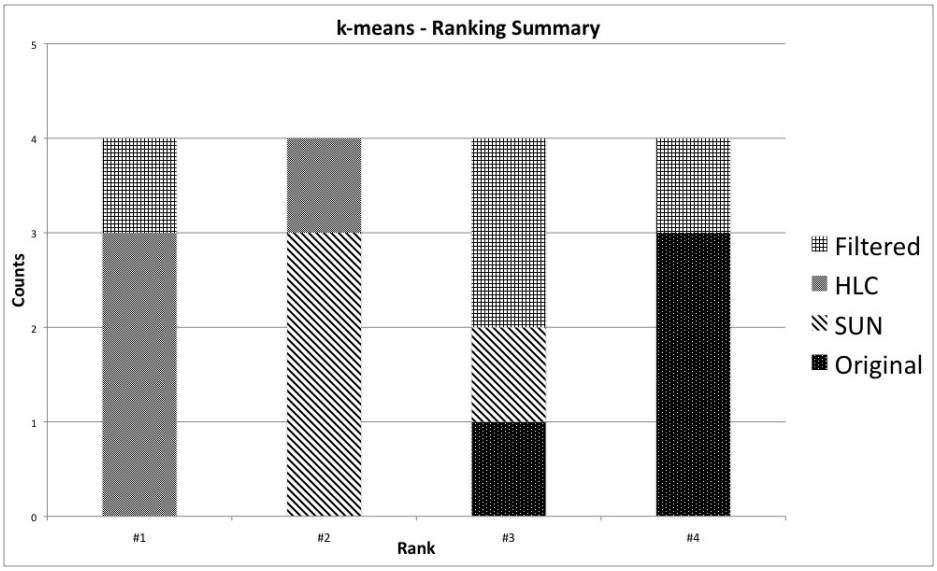
The performance of the methods was similar for the unsupervised classifier, except in this case, only the second metric needs to be considered for the ranking of methods. Due to the simplicity of ranking the *k-means* results (only one statistic to order), they will not be inclusively written down here (as was done for the supervised classifier). There are nonetheless a few results worth mentioning. The filtered method achieved the best result for the day scene. The tungsten HLC classmap is so close to the truth that the accuracy is the same for both classifier. The sodium scene has the single instance of the untreated image not getting the poorest result, it is the one instance where filtered worsened the classmap.

The bar charts of Figure 4.16 are a compilation of the ranks assigned to each class map based on the statistics from the previous table. The rank assigned was based on primarily the $\hat{k}$ for the Minimum-Distance classifier and the sum error was used to order classmaps with similar $\hat{k}$. The ranking of the supervised classifier was solely based on $\hat{k} * Class\%$ since it also takes the simple $\hat{k}$ into account, as was mentioned previously.

These charts (see Figure 4.16) allow to determine that all three methods will outperform the original image in almost all cases. The HLC image is the best in most cases, followed by the SUN image; the filtered image offers some improvement.



(a) Minimum Distance



(b) *k-means*

Figure 4.16: Compilated ranking of each image for both classifier based on the statistics of Table 4.1.

4.3 Discussion

While the majority of the simple scene consists of grass pixels, there are a good number of brick pixels and very few color panel pixels. So it is possible for a classifier to do very well if it handles the grass and brick pixels well. This is what happens with the 'day' scene. The original image causes the classifier to error wherever there is a shadow (most shadow pixels are made of grass), yet the statistics are good. This is simply because shadow pixels are a small portion of the image, more specifically a small portion of the grass class. The night scenes have considerably more spatial variations of illumination and the statistics are more representative of the quality of the classmaps obtained. The other images still manage to improve on the high $\hat{k}$ and reduce the number of errors for the 'day' scene.

The HLC space was calibrated for Planckian sources. Section 3.1.2 covers the sources used in the experiments and Figure 3.16 shows how the Planckian assumption is verified. It was established that the tungsten source is Planckian, the daytime (solar+sky) and moonlit sky (not the moon light) radiances are very close to Planckian. As should be expected, our images performed best when near blackbodies (BB) were found in the scene. The distinction between lunar and night sky radiance explains the low results for the 'moon' scene (non-BB source with BB shadows) and the good results for the 'tungsten' scene (BB source and shadows).

The classmaps from the 'mts' scene are particularly interesting (see Figure 4.10 and 4.15). There are two near-BB sources (tungsten and night sky) and two non-BB sources (lunar and sodium lamp). The region lit by the man-made sources is handled very well, while five panels are in error and some shadows on the grass are missed. The moon lit region is where most errors were made. Those observations make me believe the Planckian calibration is a good calibration for day scenes and for night scenes with most man-made sources but a different calibration is required for moonlit scenes.

Although the Filtered image does not improve much over the original image (especially compared to the results based on SUN and HLC images), it is important to note it is the only other image that retains the color information in a way readily interpretable by a user (colors consistent

with the original image).

4.3.1 Hyper-Log-Chromaticity Space

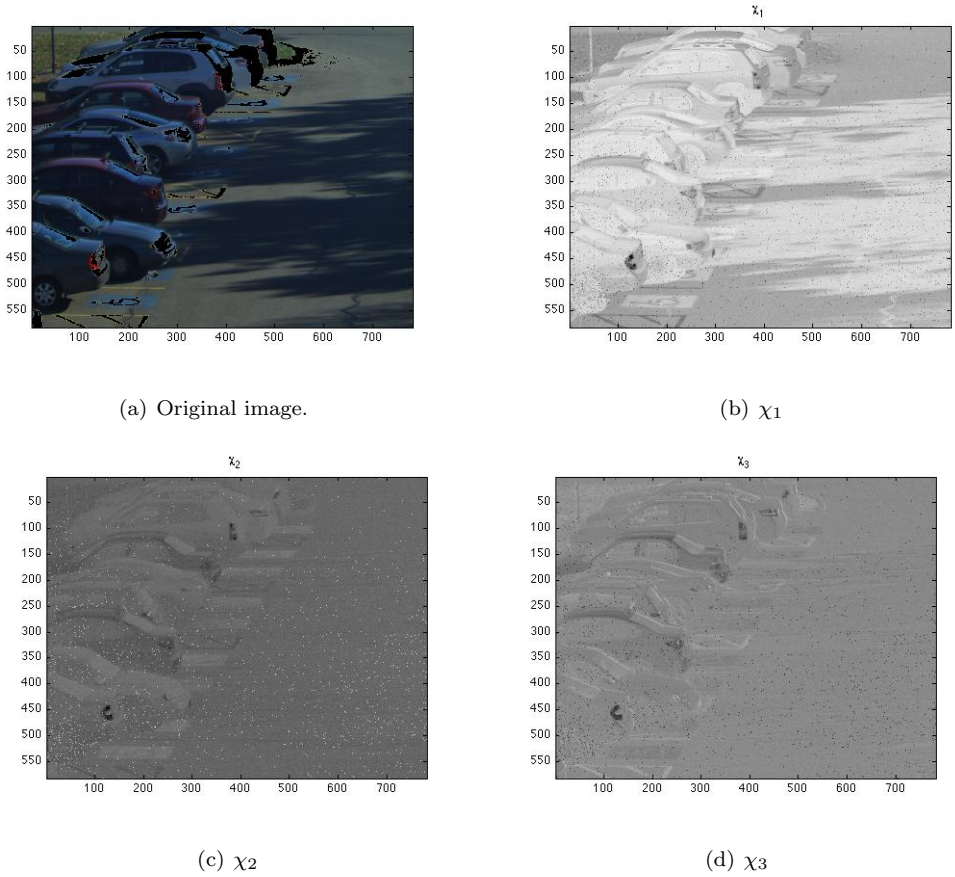
It was observed in the classmaps (see Figures 4.6(c),4.12(c)) that the two proposed algorithms have difficulty with adjacency effects and reflections. The problem comes from the basis equation that defines the HLC space, it assumes a pixel represents a surface illuminated by sources $(s + e)$. This assumption is broken when the adjacency effects or reflection are present, the appropriate representation would be:$(s_l + s_a) + e$ where s_l is the surface reflectance at the location of the pixel and s_a is the surface reflectance reflecting light onto s_l.

The problem now consists of three bi-dimensional vectors in three dimensions, e is still confined to χ_1 since an additional surface in the equation only changes the origin of the vector e, not its direction. The remaining χ will not be sufficient to separate the two separate surface reflectances. Similarly, s_a will change the origin of s_l but not its direction; s_l will be shifted towards the color of the first surface reflected.

By analyzing the location of the yellow regions in Figure 4.6(c), the shift caused by a single reflection is not sufficient to cause a significant change of s. The regions in yellow correspond to regions where more than one photon could have been reflected off a brick surface (either one photon reflected off two (or more) brick surfaces before reaching the grass or two (or more) photons reflected once off a brick surface).

Attempts to use the proposed algorithms on real images confirmed the behaviour with highly reflective surfaces (see Figure 4.17(a)). Cars that were present in the image and the pixels part of reflections were given a different HLC coordinate then the mirror-like surface. Using χ_2 and χ_3 is only sufficient to separate the illuminants from the surface reflectance, not separating surface reflectances. The dark regions in Figure 4.17(a) are caused by the method used to handle the HDR image, the highlights and dark pixels are removed from the image.

Fin (2001) [13] prove the LC space based on RGB data can only have two dimensions. The logical argument of their proof is simple. Hue, saturation and intensity are required to represent a RGB image, the LC space process removes all intensity information, so hue and saturation are left.



(a) Original image.

(b) χ_1

(c) χ_2

(d) χ_3

Figure 4.17: HLC with real images.

Table 4.2: Discrete sampling of the visible.

EigV:					
222.6556	95.8451	62.8648	16.0831	10.2114	6.2244

It was demonstrated in the presented research (see Section 3.3.2) that using the visible spectra sampled by a delta function every nanometer (the original experiments) or 20 nanometers (the 18 band images) yields at least one additional dimension with useful information. So many bands are not commonly available from sensors, so the following simple experiments were designed to take a glance at the behaviour of the HLC space with different bands.

The three experiments went through the training and calibration steps (see Section 3.3.3) with varying simulated camera responses. The first experiment consisted of touching (band center to band center = bandwidth) RECT functions:

$$RECT(x) = \begin{cases} 1 \text{ if } \mid x - a \mid \leq \frac{b}{2} \\ 0 \text{ otherwise} \end{cases} \tag{4.1}$$

where a is the parameter used to move the center of the function and b determines the width of the RECT. The RECT was ten nanometer wide ($b = 10$) for the first iteration. Each iteration increased the width by ten nanometers up to 100 nanometers (example shown in Figure 4.18). The second and third experiments consisted of Gaussians centered at the same wavelength as the RECT of the previous experiment. The second experiment used a standard deviation equal to the bandwidth and the third used a quarter of the bandwidth. The eigenvalues (EigV) of all iterations were recorded to study the dimensions of the HLC space. The top six eigenvalues (some experiments have more then six non-zero eigenvalues) are shown in the following tables (Tables 4.2, 4.3, 4.4 & 4.5).

Table 4.2 contains the six first cells from the diagonal of the eigenvalue matrix obtained when calculating the SVD of a training set of 341 bands (discrete sampling at each nanometer of the visible from 380 to 720 nm)(see Section 3.3.2). The experiments in this research did not determine the relevance of eigenvalues in HLC space, which eigenvalue represents useful information. Assuming the relation between information content and variance holds in HLC space, then the first three eigenvectors are the most representative for visualization in this case. This is not a realistic

camera response, but it sets the best case scenario for the HLC space.

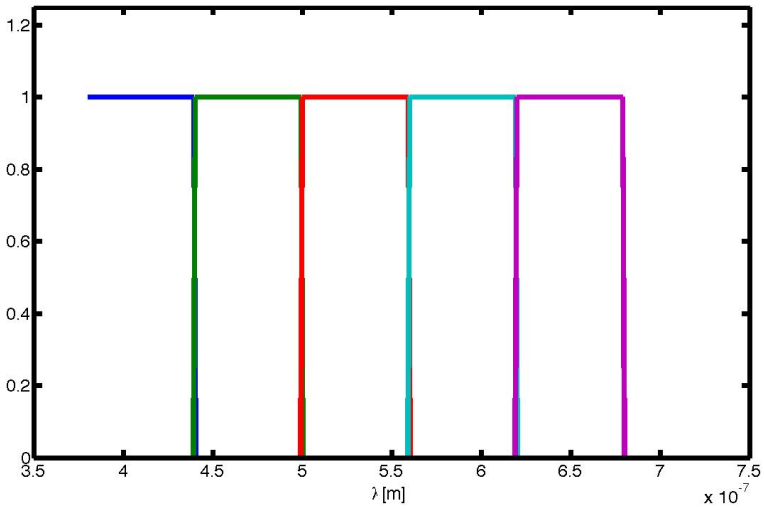


Figure 4.18: Example response of the first experiment (60nm case).

Figure 4.18 shows an example of the camera response simulated for the first experiment. RECT represents a conceptually perfect bandpass filter, there are no gap between each filter and only filters of full width were used. In the example shown, there was no room for another 60 nanometer filter in the domain, so the red portion of the spectrum is not completely sampled.

The variances of the top dimensions of the RECT response case (using 34 bands) are less then half compared to the discrete case. The variance continues to decrease as the RECT get wider and channels are removed. The three bands cases are consistent with Finlayson *et al.* work [13]. Although the relevance of eigenvalues were not determined, it was observed by looking at the 3D plots that the HLC space visibly occupies a volume when the third dimension has a greater value then 10. Based on those variances, the HLC space would require hyper-spectral data (eigenvalues of the third dimension drop under 10 with less then 11 bands).

Another to determine the useful dimension is to graph the normalized (by their sum) eigenvalues as cumulative sums (shown in Figure 4.19). With a RECT response, the top two dimensions contain 99% of the information. There is little to no difference between the 34 and 17 bands experiments, but the following experiments all show a loss of information in the higher dimensions.

Table 4.3: Study of dimensions in HLC space - RECT response.

# of bands	**width**	**EigV:**					
34	10	70.1672	30.2285	19.6986	5.0121	3.1355	1.9102
17	20	49.4708	21.2631	13.5606	3.3977	2.0675	1.2326
11	30	39.1665	17.0498	10.4962	2.5687	1.3162	0.7332
8	40	32.7713	14.3778	8.3304	1.8571	0.798	0.3751
6	50	27.3758	12.3068	6.6594	1.365	0.5054	0
5	60	24.7371	10.7042	5.3432	0.9371	0	
4	70	20.9553	9.6208	4.4441	0		
4	80	22.3018	9.1425	3.2198	0		
3	90	17.8565	7.5119	0			
3	100	18.6123	8.4076	0			

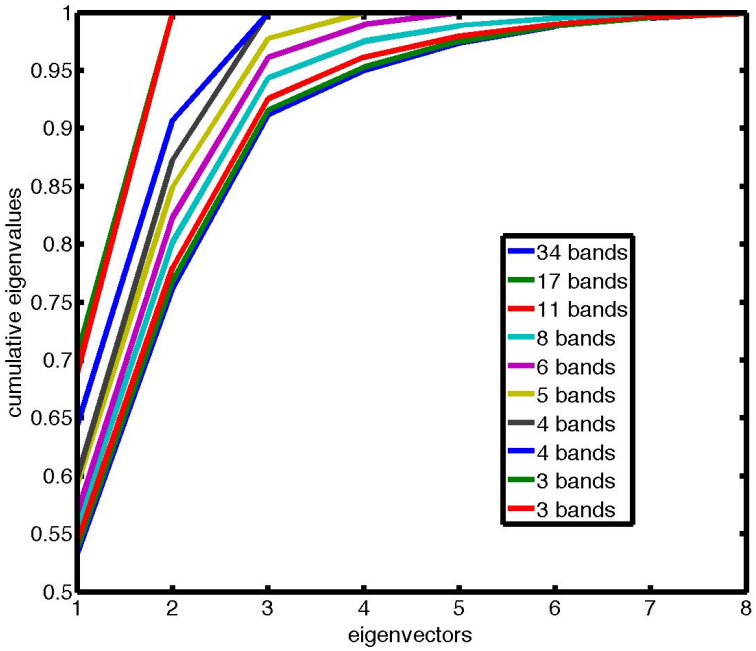


Figure 4.19: Eigenvalues for RECT experiment.

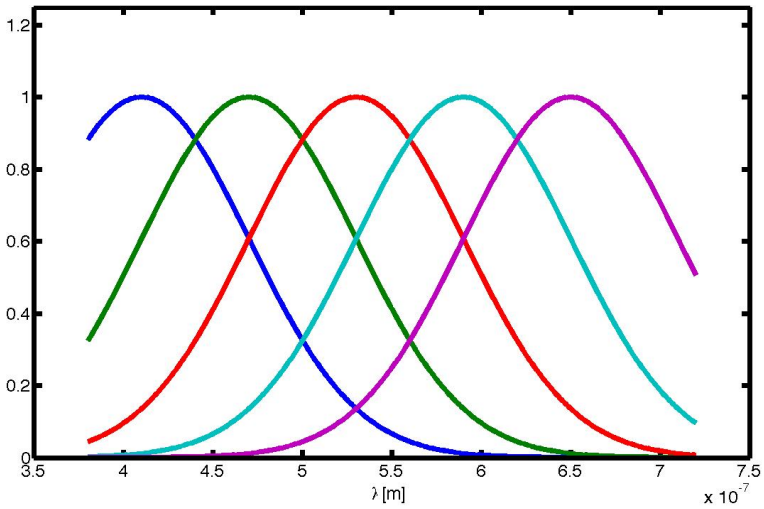


Figure 4.20: Example response of the second experiment (60nm case).

Figure 4.20 is an example of the camera response from the second experiment. The standard deviation of the Gaussian (see Equation 2.19) is equal to the bandwidth. The center of each Gaussian coincides with the center of the corresponding RECT experiment. The Gaussians are separated by a bandwidth (60 nm peak to peak in this example).

The loss in variance (of the 10 nm bands) compared to the discrete sampling case is similar to the 34 RECT bands experiments. The loss of information has the bandwidth increase and the number of bands decreases is more pronounced then the RECt experiment. The 10 and 20 nm case are the only two cases with variances above 10 in the third dimension. The second dimension drops below 5 after the 60 nm case which could mean there is too little information to even use the LC space.

Figure 4.21 confirms the wide Gaussian response is less compatible to the HLC space. The higher dimension lose more information with less bands. The slope between the second and third dimension of each experiments moves closer to zero (closer to not adding information). This was not the case with the RECT response where the slope was almost constant until it failed (at the wider four band experiment).

Table 4.4: Study of dimensions in HLC space - Gaussian response #1.

std. dev.	**EigV:**					
10	69.6569	30.0341	19.1798	4.1024	3.8726	1.4649
20	48.0516	20.4036	11.1569	2.6622	1.6042	0.6569
30	37.2161	15.1514	6.565	1.5656	0.6733	0.3459
40	29.9476	11.4006	3.6435	1.0904	0.3552	0.1471
50	23.8631	8.0834	1.9013	0.6074	0.1498	0
60	20.0171	6.0628	0.9711	0.3856	0	
70	15.9315	3.9709	0.4362	0		
80	14.5002	4.2248	0.3358	0		
90	10.7145	1.956	0			
100	9.5441	2.1041	0			

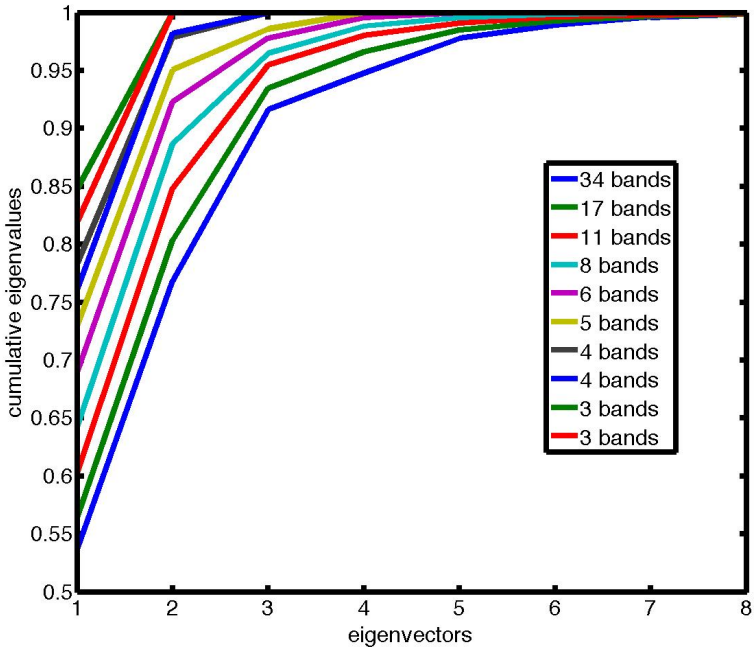


Figure 4.21: Eigenvalues for the wide Gaussian experiment.

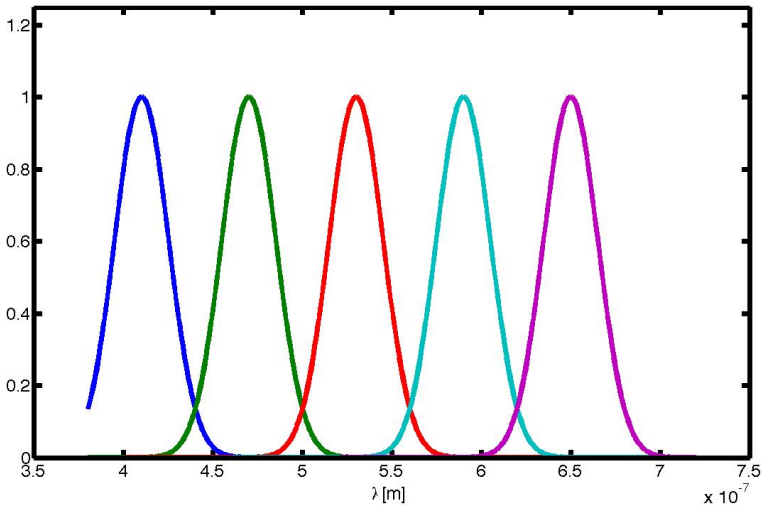


Figure 4.22: Example response of the third experiment (60nm case).

Figure 4.22 is an example of the simulated camera response from the third experiment. The standard deviation of the Gaussian is a quarter of the bandwidth and Gaussian center is separated by a bandwidth. The center of the Gaussians match the corresponding previous experiments, but the Gaussians are thinner then the previous experiment.

The results from the third experiment are overall slightly higher then the RECT experiment, but not significantly enough to consider one better then the other. The shape of each response is significantly different between all three experiments, but it would seem the overlap between responses is the most critical factor here. The majority of the responses from the second experiments overlap one another and it has the worst results. There are no overlap for the RECT experiment, the overlap is very small in the last experiment and they both have similar results.

Figure 4.23 is almost identical to Figure 4.19.

Those experiment certainly highlight the importance of the camera response in HLC space, but it is also critical to remember that the training set used in these experiments is less then ideal (see Section 3.3.3). The variance of each dimension is definitely influenced by the nature of the spectrum used. For instance, it was shown that the color patches used had higher energy returns in the red wavelengths. The slightly lower results of the RECT response experiments could be explained

Table 4.5: Study of dimensions in HLC space - Gaussian response #2.

4*std. dev.	**EigV:**					
10	70.0796	30.226	19.6021	5.0124	3.1742	1.9094
20	49.4314	21.2997	13.5686	3.4249	2.1556	1.254
30	39.2165	17.116	10.6012	2.6518	1.3722	0.7502
40	32.8349	14.4528	8.4702	1.9683	0.8314	0.3427
50	27.4446	12.383	6.7962	1.4931	0.4775	0
60	24.8045	10.8868	5.5975	1.1401	0	
70	21.1397	9.7478	4.6308	0		
80	22.2728	9.3595	3.498	0		
90	17.8146	7.7057	0			
100	18.7802	8.8759	0			

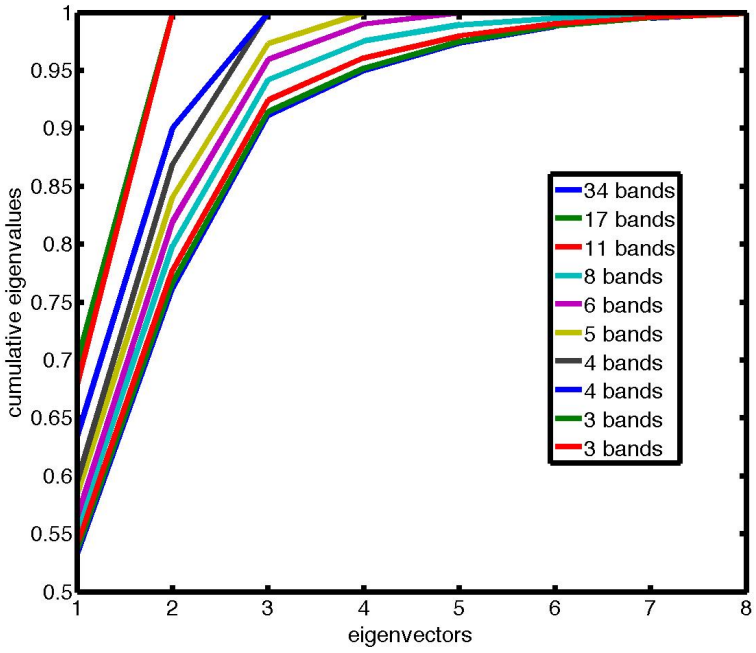


Figure 4.23: Eigenvalues for the thin Gaussian experiment.

by the portion of the red spectrum that was not sampled. Those are not in anyway conclusive, but they do hint at the lack of potential of using the HLC space with less than hyper-spectral data.

The position of samples under non-Planckian sources do not line up with Planckian sources. But their position in HLC space does not seem random. They translate in the approximate direction of the color associated with their Correlated Color Temperature (CCT). The CCT is the temperature of the Planckian radiator whose perceived color most closely resembles that of a given stimulus at the same brightness and under specified viewing conditions[1]. (The WP of a fluorescent with a CCT of 4100K which is yellow-green will be found in that color's direction relative to the WP of the Planckian sources.) Since the first dimension in HLC space (after calibration) is closely related to the illumination by design, it was tested for correlation with the sources CCT. There was none found, neither with the other dimensions or combination of the top three dimension.

Although color constancy is only achieved for the sources for which it was calibrated, the coordinates of any surface illuminated by non-Planckian sources will match approximately the equivalent under a blackbody radiator. For example a red surface under a blue light will get the same coordinates then a blue surface under a red light. Colors that look the same are getting the same coordinates. This is why the night scene were partially successful and day scene's success was dependant on surface color and not illuminants. The day scenes have near Planckian sources, the HLC space used is calibrated for those and they are handled very well. The simulated sun and sky using MODTRAN end up slightly shifted from Planckian sources, but they shift in the same direction and distance. They are nullified by the HLC space. The low results with night scenes is due to the coexistence of non-Planckian and Planckian sources; the algorithm is capable of equalizing all pixels under a single illuminant and will have a preference for the more intense source when two illuminants are present in a pixel.

The LC space shares the liabilities most luminance-chromaticity space have, neutral colors are problematic. White, black and all shades of grey have the same coordinates. When the algorithm uses cartesian coordinates instead of polar, it could confuse a light blue with a light red (going over the center). However even the polar coordinates can not prevent the algorithm from turning black

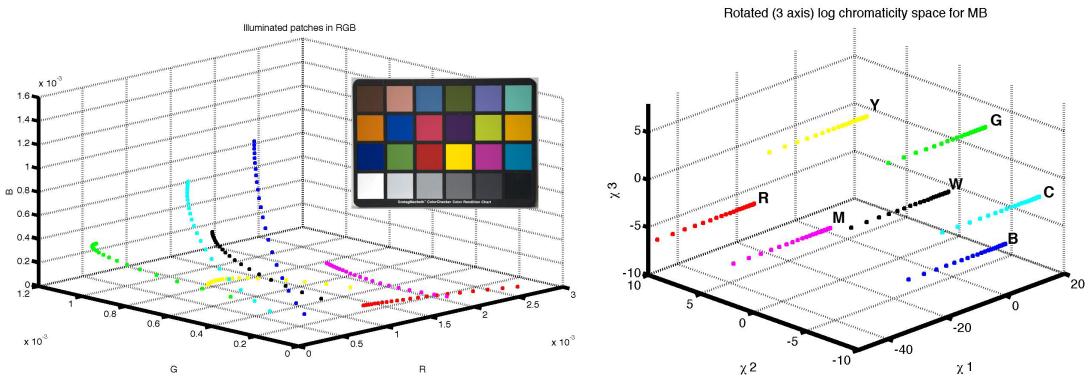
[1] http://www.cie.co.at/publ/abst/17-4-89.html

or grey pixels to white. This means the presented algorithms are less likely to error with a colorful scene then a color neutral scene. The results with Megascene2 demonstrate this point. The scene is based on a industrial plant in a near desert environment and consists mostly of variations of concrete and sand. The algorithm processes object that are spatially isolated, but starts merging grey and sandy surfaces based on which reflects more light. This explains the accuracy obtained with this scene.

There are definitive advantages to the Hyper-Log-Chromaticity space over standard luminance-chromaticity spaces and Fin (2001) version (LC). Luminance-chromaticity coordinates are derived from the scene instead of the camera so images cannot be directly compared. Either LC space are camera dependant instead of scene dependent. This means a blue tile imaged in a grass field will not have the same HSI coordinates as the same blue tile on concrete. This is not the case in LC space. Especially under the illuminants for which a space is calibrated, the blue tile will have the same chromaticity coordinates for both scenes. This should be particularly interesting for recognition applications: the ability to track colored objects through different illumination condition, environments and captures.

The advantage over luminance-chromaticity space is not limited to Planckian sources. Even without calibrating the transform for a particular space. The LC space has a rigid structure for colors under every source as is shown in Figure 3.15. The relative position of all color samples under an illuminant is constant for all illuminants. As it was demonstrated in Section 3.3.2, all sources in HLC space seem to have their own surface, exact replicas and parallel. A change of source represents a translation in the HLC space. The calibration step is the equivalent of finding a point of view to line up the subspace of different sources. If the illumination of a pixel is known, then its coordinate in HLC space can be corrected by a translation operation to correspond to a different source. Theoretically, the HLC is equivalent to a traditional hue-saturation representation if it is not calibrated for the sources present.

Luminance-chromaticity space is completely oblivious to the changes in illumination, it does nothing to differentiate the surface reflectance and the emitted spectrum. The LC space takes a step in the right direction; Finlayson's results are limited to dual source images (day scenes = sun+sky and sky (shadows)) and only one illuminant invariant dimension. The sun and sky are fairly close in color temperature so their coordinates are not far apart in the second dimension (the illuminant variant dimension). If an image has a surface lit on one side by a tungsten lamp and on



(a) RGB space (b) HLC space

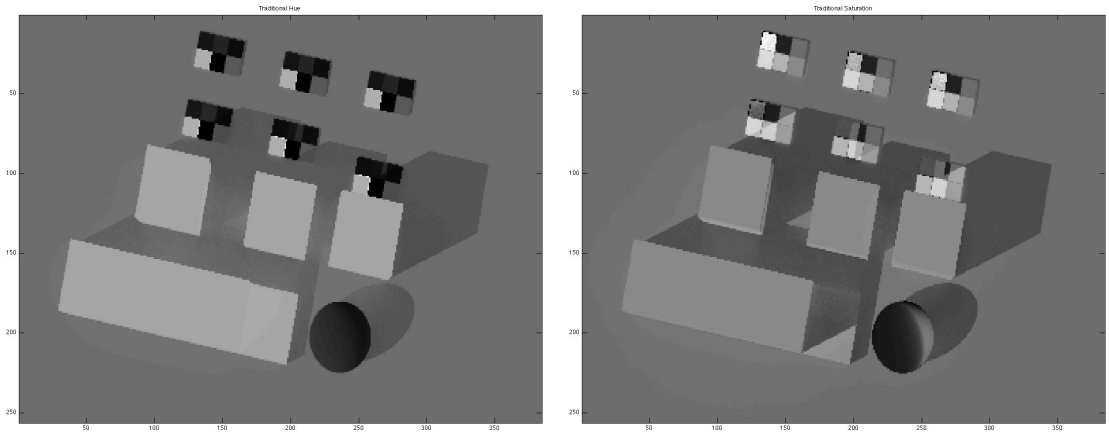
Figure 4.24: Comparison or RGB coordinates versus HLC coordinates.

the other by the sun-sky, Finlayson's space would give the same coordinate in the first dimension but something completely different in the second. Fin's (2001) color space will get decent results for blackbody radiators if they are of similar temperature in the two dimension case. The single dimension case will work if the scene does not contain colors that cause confusion. The HLC space on the other hand, will work for any blackbody within the training range and cannot be confused.

A simple way to visualize the advantage of the HLC space over an RGB space, is to show a training set in both space side by side. The training set is hyperspectral, so the RGB set was created by simulating an RGB camera response. A Gaussian with a standard deviation of 25 nm was used for all three channels. The response for the red channel (R) was centered on 650 nm, the green channel (G) on 550 nm and the blue channel (B) on 450 nm. The result is shown in Figure 4.24(a) next to the same training set in uncalibrated HLC space.

The color samples in RGB seem to trace the same curve but all in their own direction while the samples are parallel in HLC space. There is no way to separate surface reflectance from illumination in RGB space.

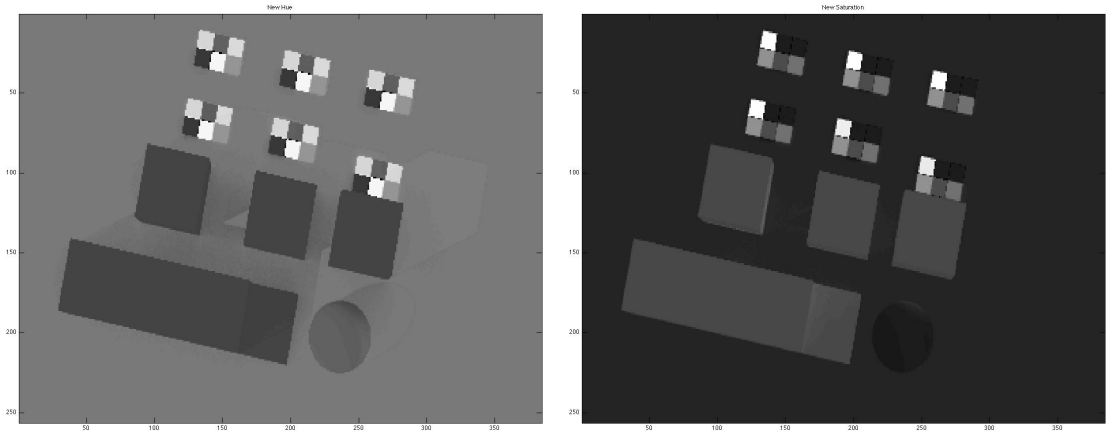
To show the advantage of the HLC space over a traditional luminance-chromaticity space, a polar version transform of the HLC was created. Figure 4.25 shows the hue and saturation channels from the traditional HSV transform described in Section 2.3 of the simple simulated day scene.



(a) Hue (b) Saturation

Figure 4.25: Representations of the simple simulated day scene (Figure 3.1(c)) in traditional hue and saturation.

Figure 4.26 shows the hue and saturation based on χ_2 and χ_3 from the same scene.



(a) HLC Hue (b) HLC Saturation

Figure 4.26: Representations of the simple simulated day scene (Figure 3.1(c)) in hue and saturation from HLC space.

The shadow regions are easily identifiable in both traditional images. Even though the intensity is not considered in those representations, the light falloff on the sphere is visible in the hue channel and somewhat visible in the saturation channel. There are also different value in both channels

for the grass pixels around the blocks (sort of halos). All those variations can cause a segmentation/classification algorithm to error. The IEF used on its own is an example of application that will error using a luminance-chromaticity space.

The HLC hue and saturation images are not completely devoid of undesired variations, but there are less of them. The sphere is approximately divided in three regions of slight variations (compared to shadow/well lit pixels from the traditional images), what would seem a direct sunlight, glancing sunlight and shadow division. The panels seem unaffected by any variation, which bodes well for a recognition application. The blocks and grass have small variations between their well lit and shadow regions. The regions where pixels could have been reflected off another surface before hitting the grass (matches the error regions in the classmaps) are also visible. Since the variations are smaller in the HLC representation, an algorithm can compensate (by adapting the standard deviations in the presented research) without risking gross errors (compensation for slight variations would cause errors with slightly different surfaces (close to neutral mostly), while compensation for large variations would cause different colors to be considered similar).

4.3.2 Illumination Equalization Filter

The early research consisted of the filter alone without the advantages of the HLC space (see Section 3.4). It was able to reduce illumination effects such as light falloff, but had an undesirable effect when attempting to remove shadows. To be made to modify shadows, the parameters values had to be very wide which caused it to blur unwanted regions. Even then it would not compensate for the color of shadows. In a day scene, a corrected shadow would appear as a bright blue region instead of dark blue. Results using only the filter are not presented because on its own its capacity for illumination equalization is limited to intensity.

Of course the performance of the filter is parameter dependent. The size of the kernel affects the run time (bigger = longer). The kernel size and the spatial standard deviation protect edges. A simple scene could be run with a big kernel and standard deviation without loosing edges and would require fewer filter passes (each run would be longer, but the filter is run fewer times). For a complex scene, the filter must be kept small and run many times, failure to do so would blur

the image. The discriminant function standard deviation controls what is to be considered evenly lit, it relates to the image's dynamic range. If σ_u is too big, it will not correct many illumination effects, too small and it will try to correct regions of equal illumination for their small differences (noise). The cutoffs (SL and HL) are also related to the image's dynamic range; they are only required when deep shadows (signal-to-noise and information too low) or highlights (blown pixel, low information) are present in an image. In the event that a dark surface is wrongfully brightened, SL can be used to protect this surface. All the results presented are the best parameter combination found for each image.

The visual impact of every pass of the IEF varies because the Euclidian distance is involved in the calculation of the weight of a neighbouring pixel. Big differences will create drastic changes in the output image while small differences will be almost indiscernible (and will require multiple passes to be corrected). A shadow pixel might look like a lit pixel after just one pass, but require multiple passes to be equal to its well lit neighbour. The filter does not fix pixels without a well lit pixel in their neighbourhood.

The filter will not necessarily equalize all surfaces. This was demonstrated in Figure 4.1 and 4.2. The panels entirely in the shadow are not corrected to look like their well lit equivalent. Of course in this particular example, the kernel of the filter could have been made big enough to include the nearest well lit panel. This goes back to scene dependent parameters, the parameters were selected to be representative of a real scene. The basic assumption of the IEF is that it is reasonable to blur the illumination of close neighbours. Not that errors cannot be made amongst close neighbours, it is simply less likely to occur with smaller neighbourhoods (spatial resolution and scene clutter consideration implied).

Another interesting observation made possible by the 'mts' scene is that the IEF has a preference for the most powerful illuminant. This is not surprising as its discriminant function is designed to favor brighter pixels. This is observable by the tungsten lit area of the class map being much larger than on the original image. This settles the question of what happens when more than one illuminant lights a pixel. On the other hand, the behaviour of the algorithm when pixels are a mix of materials is unknown.

This preference for brighter pixels could be problematic for noisy images. The standard deviation of the discriminant function σ_u does allow for some noise in the image since it controls the flat region of the function. The standard deviation should be picked to be equal or greater than

the noise present in the image. Failure to do so would allow pixels brighter due to noise to be dominant in the output image. On the other hand, if an image is very noisy, a large standard deviation would prevent slight intensity variations (like light falloff) to be corrected. In the best scenario, a smooth transition of intensity becomes a stair transition; in the worst case, it is not affected at all. The IEF is not well balanced, it has a preference for brighter pixels by design which makes it sensitive to noise. σ_u should be selected to take noise into consideration or the image could be passed through the original bilateral filter (see Section 2.6) to reduce the noise prior to using the IEF.

4.3.3 HLC and IEF

The three standard deviations controlling the similarity filter can be fairly big for a colourful scene under near Planckian sources; the further you are from that case, the smaller they need to be (especially σ_{WP}). If they are too big for the scene, some surfaces might end up overtaking others. If they are too small, some illumination effects won't be corrected. The selection of the parameters will change the performance of the algorithm and its runtime.

There are also additional consideration when multiple iterations are used: when to recalculate the HLC image, when to stop iterating, how to measure the change in the image. Because of the spatial limitation of the filter, the amount of change in the image may vary a lot between each iteration. It depends on the image. If change is the difference between the output image and the previous image, then the change can not be expected to continuously decrease with each iteration. Neither would it be wise to stop iterating after a single low change iteration because there is no way to determine when a well lit material will start influencing its shadow counterpart as the algorithm progresses.

Since the IEF uses the Euclidian distance to measure similarity-discriminate, pixels of the same material that are lit most differently are corrected more aggressively. The similarity measure is based on the HLC image. As the algorithm progresses spatially with each iteration through the image, updating the HLC image can help image regions to stabilize or be corrected more quickly. For instance, the Figures 4.26 are the HLC images based on the input image (Figure 4.1) and there are still small variations identifiable in the HLC image. As the output image approaches the desired result, the HLC image from the output will have less of these variations. On the other

hand, the HLC transform is fairly insensitive to variations in illumination so it would not be useful to recalculate the HLC image after each iteration (waist of computation). The 'Filtered' images shown in the previous section (see Section 4.1) had there HLC image recalculated every ten iterations.

The HLC space does the majority of the work as far as removing illumination effects, but the filter provides a corrected color image. Fin present their own shadow removal and back transform to a shadow free image in a number of papers. Their method requires a strong cast shadow (edge based) [11,12,14] and is demonstrated with Planckian sources. In this respect, the present research surpasses their technique as it will work with various types of sources and it can eliminate penumbrae as well as umbrae (hard edges). The IEF is a simpler method to obtain a shadow free RGB image from an LC space, but it requires multiples iterations (while Fin(2004) is a complex single pass operation).

Although it does improve in areas over other algorithms, the presented research is not the perfect solution. The sought after end result of the algorithm would be devoid of texture and structural/perspective information (usually observable by shadows or variation in illumination). Of course, this will only be achieved if all shadow materials have a well lit neighbour, so the presented algorithm has a built-in limitation. The tungsten filtered result is a good example of this. There is no way to identify the original shape of the original structures in the image. The sphere is now a circle, the blocs could be the result of a variety of different structure. This structural information is the result of interpreting the edges and all edges within a given material are removed.

From few attempts with real images (not presented here), the present algorithm has issues with reflections, but further testing is required. It is not surprising since this version of the HLC is confused by reflections off multiple surfaces. This is less then ideal for traffic monitoring because many vehicles have reflective finishes. The algorithm will still eliminate errors due to shadows but it would not prevent reflections from being identified as independent objects.

4.3.4 Summary

The following figure (Figure 4.27) is the simplified color-coded representation of this summary. The color code is attributed in function of the ability of the method to handle shadows and light falloff given the generic types of scene. Red means the method cannot handle either illumination effect while green is the opposite. The IEF is given orange because it will handle light falloff in any given scene, but never shadows if used on its own. The HLC is calibrated by default for Planckian sources, but it can be calibrated for other sources. Better results are obtained if the right calibration is used, which is why the yellow code was attributed: the method can if properly calibrated.

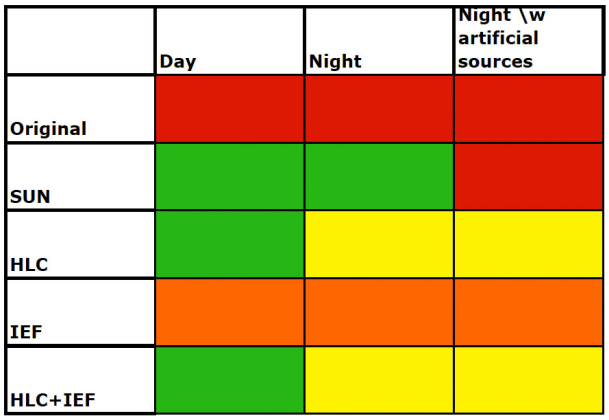


Figure 4.27: GO-NOGO chart for the presented research.

Overall the presented research has made some progress over the current solutions.

Chapter 5

Future Work

The work presented in this research is far from final. There are a number of aspects of this research that could be better understood and benefit from additional experimenting. This chapter covers areas that should be studied and ideas that have some potential. It is separated between aspects of the HLC and those of the IEF.

5.1 HLC

Finlayson's work was discovered late in the research for illumination equalization. The discovery of that third dimension that allows for the separation of the temperature of a blackbody radiator from the color of a surface is clearly instrumental in the successes of the proposed algorithms. There are many unanswered question in regards to the hyper-log-chromaticity space. Here is the list as seen by the author of this research:

- Define the conditions necessary for the third dimension to exist; a study of the: number of bands, bandwidth, band separation, band types, quantization and their effect on the HLC space; which sensors can benefit from the HLC space.

- With more bands, the eigenvalues get rather significant for the other dimensions ($\chi_4, \chi_5, etc.$); what information do they carry? Assuming colors cannot be described with less then two dimensions, then the surface reflectance requires two dimensions and two other dimensions for

the color of the source, for a total of four dimensions. As it was explained in the discussion (see Section 4.3.1), all reflections in the photon's path could be observed in HLC space, would the additional dimensions allow to separate the surface reflectance from the precedent reflections?

- Extending the study of the space beyond the visual spectrum. Is there any benefit in the near-infrared or further? Theoretically, Fin (2001) equation hold as long as Wien's approximation is valid (wavelength much shorter then the temperature times Planck's constant). At the very least it would be interesting to verify if the hyperspectral algorithms can benefit from having the visible portion of the spectrum replaced by the two HLC bands.

- Using quadratic entropy instead of Shannon's definition for the entropy minimization calibration method.

- The sources are well studied and are relatively limited in comparison to possible surface reflectances. The validity of the HLC space is dependent of the quality of the training set. The MacBeth color patches are very strong in the reds and it shows in HLC space. Color patches could be designed to be more evenly distributed in the color space.

- The HLC space eliminates all illumination effects, a back transform to RGB could perform better than using the IEF or Fin (2006) method.

- Studying the polar format further: how to choose the neutral coordinates, what is the domain of r, could it replace luminance-chromaticity color space in algorithms without requiring algorithms to be modified.

- The presented research only uses two of the three dimensions known to contain information. The first dimension could be used to distinguish the source lighting each pixels. Further characterization of the HLC space, especially the exact shape of the surface occupied by each source subspace, could enable the ability to reliably determine the illumination of each pixel.

- This research did not study the impact of noise, dynamic range and other sensor specifications have on the HLC space. Understanding the effects of exposure settings (shutter speed and aperture) on the HLC transform would be required to use the HLC space with consumer cameras.

5.2 IEF

As far as the algorithm (that merges the HLC and IEF) and the filter are concerned, most of the future work would focus on their implementation.

- smart-tiling; tiling the image in function of tile content and change brought by the filter.
- automated parameter selection.
- coded for GPU architecture.
- adaptive multiple-run: runs the filter on a tile until a desired result, when to recalculate the HLC image.
- optimization of the code (reduce memory usage, least consuming operations, etc.)
- including the processes to correct every part of an image's dynamic range (other discriminant functions).
- studying the algorithm on real images.

Chapter 6

Conclusion

A novel algorithm was presented which can improve computer vision applications by removing some variations due to illumination. This algorithm consists of two major parts: (1) a modified bilateral filter that is responsible for creating the image product and (2) a novel illuminant invariant color space. New characteristics were shown for the LC space based on RGB data initially presented by Finlayson and Drew (2001) [13]: (1) by studying its behaviour with non-Planckian sources, and (2) by extending the LC space to hyperspectral data. The HLC space is camera dependant and can be calibrated to compensate for illuminants found in a scene.

The HLC space is better suited for computer vision compared to other luminance-chromaticity spaces (i.e. HSI, HSL). In the event the camera response does not provide sufficient information for the third dimension to exist, the HLC reverts to the LC space. If the HLC space is used with sources for which it was not calibrated, then it behaves like any chromaticity space. In the best scenario using the HLC space separates the illumination and surface reflectance information; in the worst scenario, something similar to any other chromaticity space is obtained (something to gain, nothing to lose).

The IEF could be used on its own to remove falloff effects, but it was used as a mean to create a shadow free image based on the HLC space. The output of the algorithm (combining the HLC and IEF) removes most illuminations effects. This means that textures and structural/perspective information is loss in the image, but materials are free of changes due to illumination. The algorithm is suited for classification, segmentation and pattern recognition (based on surface

reflectance) applications where the objective is to properly identify the material (not its shape or position). It was demonstrated that it can improve the accuracy of two known classifiers.

Bibliography

[1] A. Berk, L.S. Bernstein, and D.C. Robertson. Modtran: A moderate resolution model for lowtran 7. *Air Force Geophysics Laboratory Technical Report, Hanscom AFB (MA)*, GL-TR-89-0122, 1989.

[2] Andrea Prati, Ivana Mikic, Mohan M. Trivedi, and Rita Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 25(7):918–923, JULY 2003.

[3] G. Buchsbaum. A spatial processor model for object colour perception. *J. Franklin Inst.*, 310:1–26, 1980.

[4] B.V. Funt, V. Cardei, and K. Barnard. Learning color constancy. *Proc. Fourth Color Imaging Conf.*, pages 58–60, NOVEMBER 1996.

[5] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Proceedings of the 1998 International Conference on Computer Vision, Bombay, India*, pages 1–8, 1998.

[6] Chuanxu Wang and Weijuan Zhang. A robust algorithm for shadow removal of foreground detection in video surveillance. *IEEE Asia-Pacific Conference on Information Processing*, pages 422–425, 2009.

[7] Chunlin Jiao, Mantun Gao, and Yikai Shi. Color image segmentation in a novel dynamic color space. *Proceedings of the7th World Congress on Intelligent Control and Automation*, pages 1–6, JUNE 2008.

[8] D.H. Brainard and W.T. Freeman. Bayesian color constancy. *J. Optical Soc. Am.*, 14(7):1393–1411, 1997.

[9] G.D. Finlayson. Color in perspective. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(10):1034–1038, OCTOBER 1996.

[10] D.A. Forsyth. A novel algorithm for colour constancy. *Int'l J. Computer Vision*, 6(1):5–36, 1990.

[11] Graham D. Finlayson, Mark S. Drew, and Cheng Lu. Intrinsic images by entropy minimization. *Springer*, pages 585–595, 2004.

[12] Graham D. Finlayson, Mark S. Drew, and Cheng Lu. Entropy minimization for shadow removal. *Int J Comput Vis*, 85(35-27):35–57, 2009.

[13] Graham D. Finlayson and Steven D. Hordley. Color constancy at a pixel. *Optical Society of America*, 18(2):253–264, FEBRUARY 2001.

[14] Graham D. Finlayson, Steven D. Hordley, Cheng Lu, and Mark S. Drew. On the removal of shadows from images. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 28(1):59–68, JANUARY 2006.

[15] Graham D. Finlayson, Steven D. Hordley, and Paul M. Hubel. Color by correlation: A simple, unifying framework for color constancy. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 23(11):1209–1221, NOVEMBER 2001.

[16] H. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer vision systems*, New York: Academic Press. In A. Hanson & E. Riseman (Eds):3–26, 1978.

[17] http://www.babelcolor.com/main_level/ColorChecker.htm#ColorChecker_data. Macbeth color checker spectra, March 2011.

[18] http://www.blender.org/. Blender, March 2011.

[19] http://www.ittvis.com/language/en us/productsservices/envi.aspx. Envi.

[20] http://www.mathworks.com/matlabcentral/fileexchange/28790-colorspace transformations. Color space conversion matlab code.

[21] http://www.ngdc.noaa.gov/dmsp/spectra.html. National oceanic and atmospheric administration - national geophysical data center.

[22] I. Mikic, P. Cosman, G. Kogut, and M.M. Trivedi. Moving shadow and object detection in traffic scenes. *Proc. Int'l Conf. Pattern Recognition*, 1:321–324, SEPTEMBER 2000.

[23] J. Stauder, R. Mech, and J. Ostermann. Detection of moving cast shadows for object segmentation. *IEEE Trans. Multimedia*, 1(1):65–76, MARCH 1999.

[24] Jean-Michel Morel, Ana Belen Petro, and Catalina Sbert. Fast implementation of color constancy algorithms. *SPIE IST*, 7241(6):1–10, 2009.

[25] Joohyun Lee, Gwanggil Jeon, and Jechang Jeong. Piecewise tone reproduction for high dynamic range imaging. *IEEE*, pages 911–918, 2009.

[26] J.R. Schott, S.D. Brown, R.V. Raqueno, H.N. Gross, and G. Robinson. An advanced synthetic image generation model and its application to multi-hyperspectral algorithm development. *Canadian Journal of Remote Sensing*, 25(2), JUNE 1999.

[27] Jwu-Sheng Hu, Tzung-Min Su, and Shr-Chi Jeng. Robust background substraction with shadow and highlight removal for indoor surveillance. *IEEE International Conference on Intelligent Robots and Systems*, OCTOBER 2006.

[28] Kobus Barnard, Lindsay Martin, Adam Coath, and Brian Funt. A comparison of computational color constancy algorithms – part ii: Experiments with image data. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 11(9):985–996, SEPTEMBER 2002.

[29] Kobus Barnard, Vlad Cardei, and Brian Funt. A comparison of computational color constancy algorithms - part i: Methodology and experiments with synthesized data. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 11(9):972–983, 2002.

[30] Kuo-Liang Chung, Yi-Ru Lin, and Yong-Huai Huang. Efficient shadow detection of color aerial images based on successive thresholding schene. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, 47(2):671–682, FEBRUARY 2009.

[31] E.H. Land. The retinex theory of color constancy. *Scientific Am.*, pages 108–129, 1977.

[32] L.T. Maloney and B.A. Wandell. Color constancy: A method for recovering surface spectral reflectance. *J. Optical Soc. Am.*, 3(1):29–33, 1986.

[33] M.M. D'Xmura and G. Iverson. Probabilistic color constancy. *Geometric Representations of Perceptual Phenomena: Papers in Honor of Tarow Indow's 70th Birthday*, 1994.

[34] Noriko Ide, Woonghee Lee, Nana Akahane, and Shigetoshi Sugawa. A wide DR and linear response CMOS image sensor with three photocurrent integrations in photodiodes, lateral overflow capacitors, and column capacitors. *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, 43(7):1577–1587, JULY 2008.

[35] Pankaj Kumar, Kuntal Sengupta, and Adrian Lee. A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system. *The IEEE 5th International Conference on Intelligent Transportation Systems*, pages 100–105, NOVEMBER 2002.

[36] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, and A. Prati. The sakbot system for moving object detection and tracking. *Video-Based Surveillance Systems–Computer Vision and Distributed Processing*, pages 145–157, 2001.

[37] W.M. Richard. Automated detection of effective scene illuminant chromaticity from specular highlights in digital images. Master's thesis, Center of Imaging Science, Rochester Inst. Technol., Rochester NY, 1995.

[38] G. Sapiro. Color and illuminant voting. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10:210–218, 1985.

[39] John R. Schott. *Remote Sensing: The imaging chain approach.* Oxford University Press, second edition, 2007.

[40] S.A. Shafer. Using color to separate reflection components. *Color Res. Applicat.*, 10:210–218, 1985.

[41] T. Horprasert, D. Hardwood, and L.S. Davis. A statistical approach for real-time robust background substraction and shadow detection. *Proc. IEEE Int'l Conf. Computer Vision '99 FRAME-RATE Workshop*, 1999.

[42] Brill Worthey. Heuristic analysis of von kries color constancy. *J. Op. Soc. Am.*, 3:1708–1712, 1986.

Appendix A

Reproduction of "Color constancy at a pixel" - Matlab Code

```
%Author: Samuel Brisebois
%Started Date: 9 December 2010

% This function was written to reproduce Finlayson's results published
% in his 2001 paper in J. Opt. to verify my understanding of his work. Also
% created figures using Planck instead of Wien to see the differences.
clc; clear all; close all;


% General Parameters
%--------------------------------------------------------------------------
lambda = 380:1:720;lambda = lambda.*1E-9;%wavelength in [m] limited to visible
T = 2500:500:10000;%temperature range in [K]
R = 610*1E-9;%Wavelength passed by red filter [m]
G = 540*1E-9;%Wavelength passed by green filter [m]
B = 450*1E-9;%Wavelength passed by blue filter [m]
sigma_f = 25*1E-9;%Filter standard deviation
sv = 1E-12;%small value to replace 0
transR = 0.5;%red filter transmission
```

```
transG = 0.5;%red filter transmission
transB = 0.5;%red filter transmission

% General Constants
%--------------------------------------------------------------------------
k = 1.381E-23;%Boltzmann constant [J/K]
h = 6.626068E-34;%Planck constant [Js]
c = 299792458;%Speed of light in the void [m/s]

%--------------------------------------------------------------------------
%--------------------------------------------------------------------------

% Creation of Planck and Wien sources
%--------------------------------------------------------------------------
%Wien's approximation assumes -1 can be dumped as long as hv >> kT.
%Normalized by the total power under the curves.
M = zeros(length(T),length(lambda));%Planck's Law
W = zeros(length(T),length(lambda));%Wien's approximation
for i = 1:1:length(T)
    M(i,:) = (2.*pi.*h.*c^2)./(lambda.^5.*(exp((h.*c)./(lambda.*k.*T(i)))-1));
    M(i,:) = M(i,:)./sum(M(i,:),2);
    W(i,:) = (2.*pi.*h.*c^2)./(lambda.^5.*exp((h.*c)./(lambda.*k.*T(i))));
    W(i,:) = W(i,:)./sum(W(i,:),2);
end;


%Fig. 1
%--------------------------------------------------------------------------
figure; plot(lambda,M(find(T == 2500),:),'k',lambda,W(find(T == 2500),:),'r-');
title('Fig 1: Normalized 2500 [K] blackbody radiator');
xlabel('Wavelength \lambda [m]');
legend('Plancks Law','Wiens approximation');
%--------------------------------------------------------------------------

%Fig. 2
%--------------------------------------------------------------------------
figure; plot(lambda,M(find(T == 5500),:),'k',lambda,W(find(T == 5500),:),'r-');
title('Fig 2: Normalized 5500 [K] blackbody radiator');
```

```
xlabel('Wavelength \lambda [m]');
legend('Plancks Law','Wiens approximation');
%------------------------------------------------------------------------

%Fig. 3
%------------------------------------------------------------------------
figure; plot(lambda,M(find(T == 10000),:),'k',lambda,W(find(T == 10000),:),'r-');
title('Fig 3: Normalized 10000 [K] blackbody radiator');
xlabel('Wavelength \lambda [m]');
legend('Plancks Law','Wiens approximation');
%------------------------------------------------------------------------

%Import DIRSIG spectra from excel
warning off;
sources = xlsread('dirsig_sun_and_sky95.xls','Import');%sun,sky,sun+sky
sources(:,2:4) = sources(:,2:4).*10000.*1E-6;
daylight = interp1(sources(:,1),sources(:,4),lambda,'cubic');
daylight = daylight./sum(daylight,2);

%Fig. 4
%------------------------------------------------------------------------
figure; plot(lambda,M(find(T == 5500),:),'k',lambda,daylight,'r-');
title('Normalized 5500 [K] blackbody radiator VS normalized daylight spectra');
xlabel('Wavelength \lambda [m]');
legend('Plancks Law','DIRSIG daylight');
%------------------------------------------------------------------------

%Import MacBeth Color Checker board spectra from excel
warning off;
MacBeth = xlsread('ColorChecker95.xls','spectral_data');
MacBeth = MacBeth(2:26,3:38);%wavelength and all colors
MacBeth(1,:) = MacBeth(1,:).*1E-9;
temp = MacBeth([8,11,14,15,16,17,20],:);%Finlayson uses G,Y,W,B,P,O&R
colors = zeros(size(temp,1),length(lambda));%O,P,B,G,R,Y&W
for i = 1:1:size(colors,1)%Just keeping the same patches used by Finlayson
    colors(i,:) = interp1(MacBeth(1,:),temp(i,:),lambda,'cubic');
end;
clear temp;
```

```
%-------------------------------------------------------------------------
figure; plot(lambda,colors(1,:),'c',lambda,colors(2,:),'m',lambda,colors(3,:),'b',lambda,colors(4,:),'g',lambda,co
title('Spectra of 7 color patches');
xlabel('Wavelength \lambda [m]');
%-------------------------------------------------------------------------

%finding the indices of the desired wavelength
red = find(lambda == R); green = find(lambda == G); blue = find(lambda == B);

%Creating RGB channels based on Wien illumination and Δ camera response
redWd = colors(:,red)*W(:,red)';
greenWd = colors(:,green)*W(:,green)';
blueWd = colors(:,blue)*W(:,blue)';

%Calculating Log-Chromaticity differences.
rho1Wd = log(redWd./greenWd);
rho2Wd = log(blueWd./greenWd);

%Fig. 6
%-------------------------------------------------------------------------
figure; plot(rho1Wd(1,:),rho2Wd(1,:),'c.',rho1Wd(2,:),rho2Wd(2,:),'m.',rho1Wd(3,:),rho2Wd(3,:),'b.',rho1Wd(4,:),rh
title('Fig 6: Perfect Dirac Δ camera (Wien)');
xlabel('LN R/G'); ylabel('LN B/G');
%-------------------------------------------------------------------------

rho = [reshape(rho1Wd',[],1),reshape(rho2Wd',[],1)];

% Demeaning
%-------------------------------------------------------------------------
u = zeros(size(colors,1),2);% initialing mean matrix
for i = 1:size(colors,1)
    u(i,:) = mean(rho((i-1)*size(W,1)+1:i*size(W,1),:),1);
end;

Gamma = zeros(size(rho));% initiating Gamma
for i = 1:size(colors,1);
    Gamma((i-1)*size(W,1)+1:i*size(W,1),:) = rho((i-1)*size(W,1)+1:i*size(W,1),:) - ones(size(W,1),1)*u(i,:);
```

```
end;

% SVD
%-------------------------------------------------------------------------
[U,S,V] = svd(Gamma,0);
rhopd = rho*V; rhopd = rhopd(:,1:2);
rhopd = reshape(rhopd,size(W,1),size(rho,1)/size(W,1),[]);
rhopd = cat(3,rhopd(:,:,1)',rhopd(:,:,2)');

%Fig. 7
%-------------------------------------------------------------------------
figure; plot(rhopd(1,:,2),rhopd(1,:,1),'c.',rhopd(2,:,2),rhopd(2,:,1),'m.',rhopd(3,:,2),rhopd(3,:,1),
title('Fig 7: Rotated Perfect Dirac Δ camera (Wien)');
xlabel('\chi_2'); ylabel('\chi_1');
%-------------------------------------------------------------------------

%same as above but for Planck with dirac Δ camera response
redMd = colors(:,red)*M(:,red)';
greenMd = colors(:,green)*M(:,green)';
blueMd = colors(:,blue)*M(:,blue)';

%Calculating Log-Chromaticity differences.
rho1Md = log(redMd./greenMd);
rho2Md = log(blueMd./greenMd);

%Not part of the paper, for comparison.
%-------------------------------------------------------------------------
figure; plot(rho1Md(1,:),rho2Md(1,:),'c.',rho1Md(2,:),rho2Md(2,:),'m.',rho1Md(3,:),rho2Md(3,:),'b.',r
title('Perfect Dirac Δ camera (Planck)');
xlabel('LN R/G'); ylabel('LN B/G');
%-------------------------------------------------------------------------

% Simulated Gaussian camera response
RR = double(transR.*exp(-((lambda-R).^2)/(2*sigma_f.^2)));
GR = double(transG.*exp(-((lambda-G).^2)/(2*sigma_f.^2)));
BR = double(transB.*exp(-((lambda-B).^2)/(2*sigma_f.^2)));

% Fig. 9
```

```
%-------------------------------------------------------------------------
figure; plot(lambda,RR,'r',lambda,GR,'g',lambda,BR,'b');
title('Fig 9: Simulated camera response');
xlabel('Wavelength \lambda [m]');
%-------------------------------------------------------------------------

% Illuminating all 7 patches with all Wien sources
illuCW = zeros(size(colors,1)*size(W,1),size(lambda,2));
illuCM = illuCW;
counter = 1;
for i = 1:1:size(colors,1);
    for j = 1:1:size(W,1);
        illuCW(counter,:) = colors(i,:).*W(j,:);
        illuCM(counter,:) = colors(i,:).*M(j,:);
        counter = counter + 1;
    end;
end;
clear counter i j;

%Creating RGB channels based on Wien sources and Gaussian camera response
redWg = (illuCW*RR')'+ones(1,size(colors,1)*size(W,1)).*sv;
greenWg = (illuCW*GR')'+ones(1,size(colors,1)*size(W,1)).*sv;
blueWg = (illuCW*BR')'+ones(1,size(colors,1)*size(W,1)).*sv;

%Calculating Log-Chromaticity differences.
rho1Wg = log(redWg./greenWg); rho1Wg = reshape(rho1Wg,size(W,1),size(colors,1))';
rho2Wg = log(blueWg./greenWg); rho2Wg = reshape(rho2Wg,size(W,1),size(colors,1))';

%Fig. 10
%-------------------------------------------------------------------------
figure; plot(rho1Wg(1,:),rho2Wg(1,:),'c.',rho1Wg(2,:),rho2Wg(2,:),'m.',rho1Wg(3,:),rho2Wg(3,:),'b.',rho1Wg(4,:),rh
title('Fig 10: Simulated camera (Wien)');
xlabel('LN R/G'); ylabel('LN B/G');
%-------------------------------------------------------------------------

rho = [reshape(rho1Wg',[],1),reshape(rho2Wg',[],1)];

% Demeaning
```

```
%------------------------------------------------------------------------
u = zeros(size(colors,1),2);% initialing mean matrix
for i = 1:size(colors,1)
    u(i,:) = mean(rho((i-1)*size(W,1)+1:i*size(W,1),:),1);
end;

Gamma = zeros(size(rho));% initiating Gamma
for i = 1:size(colors,1);
    Gamma((i-1)*size(W,1)+1:i*size(W,1),:) = rho((i-1)*size(W,1)+1:i*size(W,1),:) - ones(size(W,1),1)
end;

% SVD
%------------------------------------------------------------------------
[U,S,V] = svd(Gamma,0);
rhopd = rho*V; rhopd = rhopd(:,1:2);
rhopd = reshape(rhopd,size(W,1),size(rho,1)/size(W,1),[]);
rhopd = cat(3,rhopd(:,:,1)',rhopd(:,:,2)');

%Fig. 11
%------------------------------------------------------------------------
figure; plot(rhopd(1,:,2),rhopd(1,:,1),'c.',rhopd(2,:,2),rhopd(2,:,1),'m.',rhopd(3,:,2),rhopd(3,:,1),
title('Fig 11: Rotated Gaussian camera (Wien)');
xlabel('\chi_2'); ylabel('\chi_1');
%------------------------------------------------------------------------

%same as above but for Planck sources and Gaussian camera response
redMg = (illuCM*RR')'+ones(1,size(colors,1)*size(W,1)).*sv;
greenMg = (illuCM*GR')'+ones(1,size(colors,1)*size(W,1)).*sv;
blueMg = (illuCM*BR')'+ones(1,size(colors,1)*size(W,1)).*sv;

%Calculating Log-Chromaticity Differences.
rho1Mg = log(redMg./greenMg); rho1Mg = reshape(rho1Mg,size(M,1),size(colors,1))';
rho2Mg = log(blueMg./greenMg); rho2Mg = reshape(rho2Mg,size(M,1),size(colors,1))';

%Not part of paper, for comparison.
%------------------------------------------------------------------------
figure; plot(rho1Mg(1,:),rho2Mg(1,:),'c.',rho1Mg(2,:),rho2Mg(2,:),'m.',rho1Mg(3,:),rho2Mg(3,:),'b.',r
title('Simulated camera (Planck)');
```

```
xlabel('LN R/G'); ylabel('LN B/G');
%-------------------------------------------------------------------------

%-------------------------------------------------------------------------
% This finishes Finlayson's work in 2001.
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------

% Illuminating different color patches to see where they map in LC.
%-------------------------------------------------------------------------
temp = MacBeth([14,15,16,17,18,19,20],:);%B,G,R,Y,M,C&W
colors = zeros(size(temp,1),length(lambda));
for i = 1:1:size(colors,1)
    colors(i,:) = interp1(MacBeth(1,:),temp(i,:),lambda,'cubic');
end;
clear temp;

% Illuminating all 7 patches with all Wien sources
illuCW = zeros(size(colors,1)*size(W,1),size(lambda,2));
illuCM = illuCW;
counter = 1;
for i = 1:1:size(colors,1);
    for j = 1:1:size(W,1);
        illuCW(counter,:) = colors(i,:).*W(j,:);
        illuCM(counter,:) = colors(i,:).*M(j,:);
        counter = counter + 1;
    end;
end;
clear counter i j;

%Creating RGB channels based on Planck sources and Gaussian camera response
redMg = (illuCM*RR')'+ones(1,size(colors,1)*size(W,1)).*sv;
greenMg = (illuCM*GR')'+ones(1,size(colors,1)*size(W,1)).*sv;
blueMg = (illuCM*BR')'+ones(1,size(colors,1)*size(W,1)).*sv;

%Calculating Log-Chromaticity Differences.
rho1Mg = log(redMg./greenMg); rho1Mg = reshape(rho1Mg,size(M,1),size(colors,1))';
rho2Mg = log(blueMg./greenMg); rho2Mg = reshape(rho2Mg,size(M,1),size(colors,1))';
```

```
%-------------------------------------------------------------------------
figure; plot(rho1Mg(1,:),rho2Mg(1,:),'b.',rho1Mg(2,:),rho2Mg(2,:),'g.',rho1Mg(3,:),rho2Mg(3,:),'r.',r
title('Other colors (Planck)');
xlabel('LN R/G'); ylabel('LN B/G');
%-------------------------------------------------------------------------
```

Appendix B

Training and Calibration of the Hyperspectral Log-Chromaticity (HLC) Space - Matlab Code

```
%Author: Samuel Brisebois
%Started Date: 18 May 2011

% This function is used to train and calibrate the HLC space.
% The camera response can be imported or simulated. It uses modeled
% Planckian sources and patches from the Macbeth Color chart to generate
% the training set. The calibration is automated for Planckian sources. It
% outputs the variables required by LC_Multi_Transform.m to calculate the
% coordinates in HLC space of an image.

clc; clear all; close all;

% General Parameters
%--------------------------------------------------------------------------
lambda = 380:20:720;lambda = lambda.*1E-9;%wavelength in [m] limited to visible
T = 2000:250:7000;%temperature range in [K]
```

```
sv = 1E-15;%Small value that replaces zero to prevent division by zero

% Simulated Gaussian RGB Parameters
%--------------------------------------------------------------------------
R = 650E-9; G = 550E-9; B = 450E-9;%wavelength in [m]
tR = 1; tG = 1; tB = 1;%filter transmission [0,1]
sigma_f = 25E-9;%standard deviation of the Gaussian [m]

% General Constants
%--------------------------------------------------------------------------
k = 1.381E-23;%Boltzmann constant [J/K]
h = 6.626068E-34;%Planck constant [Js]
c = 299792458;%Speed of light in the void [m/s]

%--------------------------------------------------------------------------
% Import/simulate camera response here and comment out undesired response.
%--------------------------------------------------------------------------
%each row is a band for the visible spectra, extrapolate to match the
%domain (lambda). [see the import of the Macbeth Color patches for an
%example]

%default (no camera response):
CR = double(eye(length(lambda)));%camera response

% %simulated (demo Gaussian RGB):
% CR = ones(3,length(lambda));%initaling camera response
% CR(1,:) = double(tR.*exp(-((lambda-R).^2)/(2*sigma_f.^2)));
% CR(2,:) = double(tG.*exp(-((lambda-G).^2)/(2*sigma_f.^2)));
% CR(3,:) = double(tB.*exp(-((lambda-B).^2)/(2*sigma_f.^2)));
% %--------------------------------------------------------------------------
% figure; plot(lambda,CR(1,:),'r',lambda,CR(2,:),'g',lambda,CR(3,:),'b');
% title('Fig 9: Simulated camera response');
% xlabel('Wavelength \lambda [m]');
% %--------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Planck's equation to simulate BlackBody radiators (BB)
%--------------------------------------------------------------------------
```

```
M = zeros(length(T),length(lambda));%Planck's Law
for i = 1:1:length(T)
    M(i,:) = (2.*pi.*h.*c^2)./(lambda.^5.*(exp((h.*c)./(lambda.*k.*T(i)))-1));
end;

% %-------------------------------------------------------------------------
% figure; plot(lambda,M(1,:)./sum(M(1,:),2),'r',lambda,M(floor(length(T)/4),:)./sum(M(floor(length(T)
% title('Planckian sources normalized by their power under the curve');
% legend(strcat(num2str(T(1)),' K'),strcat(num2str(T(floor(length(T)/4))),' K'),strcat(num2str(T(floo
% xlabel('Wavelength \lambda [m]');xlim([min(lambda) max(lambda)]);
% %-------------------------------------------------------------------------

sources = M;% Other sources could be included
clear M;

%Correcting possible interpolation errors
sources(find(sources < 0)) = 0;

%Normalize all sources by their power under the curve
%--------------------------------------------------------------------------
sources = sources ./ (sum(sources,2)*ones(1,size(sources,2)));


%--------------------------------------------------------------------------
% Import of color information
%--------------------------------------------------------------------------

% Import MacBeth Color Checker board spectra from excel
%--------------------------------------------------------------------------
warning off;
MacBeth = xlsread('ColorChecker95.xls','spectral_data');
MacBeth = MacBeth(2:26,3:38);%wavelength and all colors
MacBeth(1,:) = MacBeth(1,:).*1E-9;%unit conversion
SpecC = [14,15,16,17,18,19,20];%B,G,R,Y,M,C&W
temp = MacBeth(SpecC,:);
MBcolor = zeros(size(temp,1),length(lambda));
for i = 1:1:size(MBcolor,1)
    MBcolor(i,:) = interp1(MacBeth(1,:),temp(i,:),lambda,'cubic');
```

```
end;
clear temp;

%-------------------------------------------------------------------------
figure;plot(lambda,MBcolor(1,:),'b',lambda,MBcolor(2,:),'g',lambda,MBcolor(3,:),'r',lambda,MBcolor(4,:),'y',lambda
title('MacBeth color patches');xlim([min(lambda) max(lambda)]);
xlabel('Wavelength \lambda [m]');
%-------------------------------------------------------------------------

colors = MBcolor;% Other color spectra could be included
clear MBcolor;

%Normalize all color spectrums by their power under the curve
%-------------------------------------------------------------------------
colors = colors ./ (sum(colors,2)*ones(1,size(colors,2)));

%-------------------------------------------------------------------------
% Illuminating the color information (spectra)
%-------------------------------------------------------------------------
illuC = zeros(size(colors,1)*size(sources,1),size(lambda,2));
counter = 1;
for i = 1:1:size(colors,1);
    for j = 1:1:size(sources,1);
        illuC(counter,:) = colors(i,:).*sources(j,:);
        counter = counter + 1;
    end;
end;
clear counter i j;

% Taking the camera response into consideration
%-------------------------------------------------------------------------
illuC = illuC*CR';

% Replacing values of '0' by a very small value to prevent division by zero
%-------------------------------------------------------------------------
illuC = illuC+ones(size(illuC)).*sv;

% Scaling for MATLAB
```

```
illuC = illuC.*0.5E6;

%-------------------------------------------------------------------------
% Hyper-Log-Chromaticity
%-------------------------------------------------------------------------
% Using the original training method (Finlayson et al. 2001).

gm = prod(illuC,2).^(1/size(illuC,2));%Geometric Mean
rho = log(illuC./(gm*ones(1,size(illuC,2))));%rho for original method: LCD

% Demeaning
%-------------------------------------------------------------------------
u = zeros(size(colors,1),size(rho,2));% initialing mean matrix
for i = 1:size(colors,1)
    u(i,:) = mean(rho((i-1)*size(sources,1)+1:i*size(sources,1),:),1);
end;

Gamma = zeros(size(rho));% initiating Gamma
for i = 1:size(colors,1);
    Gamma((i-1)*size(sources,1)+1:i*size(sources,1),:) = rho((i-1)*size(sources,1)+1:i*size(sources,1
ones(size(sources,1),1)*u(i,:);
end;

% SVD
%-------------------------------------------------------------------------
[U,S,V] = svd(Gamma,0);%demeaned LCD
rhop = rho*V; rhop = rhop(:,1:3);

%Recording transform to create calibration worksheet for LC_Multi_Transform
save Dimension V;
At = eye(3);
save Atransform At;

rhop = reshape(rhop,size(sources,1),size(rho,1)/size(sources,1),[]);
rhop = cat(3,rhop(:,:,1)',rhop(:,:,2)',rhop(:,:,3)');

%-------------------------------------------------------------------------
figure;plot3(rhop(1,:,1),rhop(1,:,2),rhop(1,:,3),'b+',rhop(2,:,1),rhop(2,:,2),rhop(2,:,3),'g+',rhop(3
```

```
title('3D (u-SVD) log chromaticity space for MB'); xlabel('\chi 1'); ylabel('\chi 2');zlabel('\chi 3');
%-------------------------------------------------------------------------

WP = [mean(rhop(7,:,2)),mean(rhop(7,:,3))];
At = eye(3);

%Recording transform to create calibration worksheet for LC_Multi_Transform
save('Transform','V','At','WP');
```

Appendix C

Image Transform to HLC Space - Matlab Code

```
%Author: Samuel Brisebois
%Started Date: 20 January 2011

function img_o = LC_Multi_Transform_v1(img_i,transform)
%LC_Multi_Transform uses the calibration transform found by
%Multispectral_LC or Train_Calib_HLC to obtain the 3D
% Hyper-Log-Chromaticity space inspired from Finlayson's et al. work.
%
%   The calibration transforms are saved MATLAB variables and must be
%   included in the file containing LC_Multi_Transform_v1. One variable is
%   named 'Dimension' containing the eigenvectors and 'Atransform'
%   containing the 3-by-3 affine rotation transform. It ouputs
%   an image with three bands (chi1,chi2 and chi3).

%--------------------------------------------------------------------------
sz = size(img_i);
trans = load(transform);

%geometric mean of each pixel
```

```
gm = prod(img_i,3).^(1/size(img_i,3));
gm = reshape(gm,[],1);

%calculating log-chromaticity differences
rho = reshape(img_i,sz(1)*sz(2),sz(3));
rho = log(rho./(gm*ones(1,size(rho,2))));

%importing eigenvectors
%V = load ;

rhop = rho*trans.V;rhop = rhop(:,1:3);

%importing affine transform
%Atransform = load('Atransform');

rhop = rhop*trans.At;

img_o = reshape(rhop,sz(1),sz(2),3);
end
```

Appendix D

Illumination Equalization Filter (IEF) - Matlab Code

```
%Author: Samuel Brisebois
%Started: 25 October 2010

function img_o = LC_BilFiltMod(img_i,LCi,WP,ks,sigma_c,minsigma_schi1,maxsigma_schi1,maxsigma_schi2,s
%LC_BILFILTMOD is an adaptation of the bilateral-filter to perfrom
%illumination equalization. It requires the original image with its
%Hyper-Log-Chromaticity image and nine parameters. The code is included to plot the
%filters for a pixel of interest.
%   -WP is the white point for Planckian sources for the LC-Chroma transform
%   used.
%   -ks is the kernel size of the filter.
%   -sigma_c is the standard deviation of the closeness fitler (2D spatial
%   Gaussian).
%   -minsigma_shi1 is the standard deviation for the similarity filter for
%   chi1 near WP.
%   -maxsigma_schi1 is the standard deviation for the similarity filter
%   for the chi1 furthest from the WP.
%   -maxsigna_schi2 is the standard deviation for the similarity filter for
%   the chi2 furthest from the WP.
```

```
%   -sigma_u is the standard deviation for the discriminant function.
%   -HL is the highlight cutoff.
%   -SL is the shadow cutoff.
%--------------------------------------------------------------------------

img_i = double(img_i);
dims = size(img_i);

%--------------------------------------------------------------------------
% Determining the standard deviation for the similarity filter
%--------------------------------------------------------------------------

% Creating possible values for sigma_chi1
dist1 = abs(LCi(:,:,1)-ones([dims(1) dims(2)]).*WP(1,1));
D = 0:0.01:max(dist1(:));
sigma1 = interp1([min(D) max(D)],[minsigma_schi1 maxsigma_schi1],D);

% Creating possible values for sigma_chi2
dist2 = abs(LCi(:,:,2)-ones([dims(1) dims(2)]).*WP(1,2));
D = 0:0.01:max(dist2(:));
sigma2 = interp1([min(D) max(D)],[minsigma_schi1 maxsigma_schi2],D);
%--------------------------------------------------------------------------

%Preparing the images for filtering
wimg = img_i;
img_o = wimg;
wimg = Image_Unwrapping(wimg,ks);%Image_Unwrapping mirrors the edges of the image.
LCi = Image_Unwrapping(LCi,ks);

% Calculating intensity image and scaling to 32bits.
Intensity = sum(wimg,3)./dims(3);
Intensity = Intensity./max(Intensity(:)).*(2^32);

%Implementation of highlight cutoff
maxI = max(Intensity(:));
Ilimit = HL*maxI;
Intensity(Intensity > Ilimit) = 0;
```

```
%creating closeness filter based on a Gaussian function
x = -ks:1:ks;y = -ks:1:ks;
[X Y] = meshgrid(x,y);
cf = double(exp(-((X.^2+Y.^2)/(2*sigma_c.^2))));

%'for loop' creating a full band image using a modified bilateral filter to
%equalize the illumination
for i = ks+1:1:ks+dims(1)%1 row at a time
    for j = ks+1:1:ks+dims(2)%1 pixel per column at a time
        %'if' statement implements shadow cutoff
        if ¬((Intensity(i,j) ≤ SL*maxI));

            %Selection of standard deviation for similarity filter
            sigma_schi1 = sigma1(1,floor(dist1(i-ks,j-ks))+1);
            sigma_schi2 = sigma2(1,floor(dist2(i-ks,j-ks))+1);

            schi1 = exp(-((LCi(x+i,y+j,1)-LCi(i,j,1)).^2)./(2*sigma_schi1^2));%the chi1 similarity fi
            schi2 = exp(-((LCi(x+i,y+j,2)-LCi(i,j,2)).^2)./(2*sigma_schi2^2));%the chi2 similarity fi
            uf = sign((Intensity(x+i,y+j)-Intensity(i,j))).*exp(-((Intensity(x+i,y+j)-Intensity(i,j))
            csuf = cf.*schi1.*schi2.*uf;%closeness, similarity and discrimination filters combined

            %Calculating pixel value for all bands
            for k=1:1:dims(3)
                img_o(i-ks,j-ks,k) = sum(sum(wimg(x+i,y+j,k).*(csuf./sum(csuf(:)))));
            end;

% %-------------------------------------------------------------------------
% % Showing the filters of one particular pixel of the image at coordinate
% % (px,py).
% %-------------------------------------------------------------------------
%         px = 144;
%         py = 30;
%         %-----------------------------------------------------------------
%         if ((i == py+ks) && (j == px+ks))
%             figure;subplot(1,2,1);mesh(X,-Y,schi1);title(strcat('Similarity filter in \chi_1 at pos
%             subplot(1,2,2);mesh(X,-Y,schi2);title(strcat('Similarity filter in \chi_2 at position (
% % %             subplot(1,3,3);mesh(X,-Y,suf);title(strcat('Combined umbra and similarity filter at p
%             figure;subplot(1,2,1);mesh(X,-Y,uf);title(strcat('Discrimination in Intensity at positi
```

```
%             subplot(1,2,2);mesh(X,-Y,csuf);title('Complete filter')%show combined filter
%         end
% %-------------------------------------------------------------------------
        end;
    end;
end;
```